89

128er

Markt&Technik
öS 180,-/sfr 24,-/hfl 29,-/Lit 28000/dkr 100,- DM 24,-

64'er

über 15 Programme auf Diskette

DISKETTE IM HEFT

PC

128

DOS-Copy: *konvertiert C-128-Texte ins MS-DOS-Format*
Codiman: *Profi-Disk-Management mit reichlich Komfort*
MasterBasic: *51 neue Anweisungen und 25 Funktionen*
CP/M: *Starke Public-Domain-Programme (fast) zum Nulltarif*

An alle Spiele-Freaks! Das neue POWER PLAY ist da!



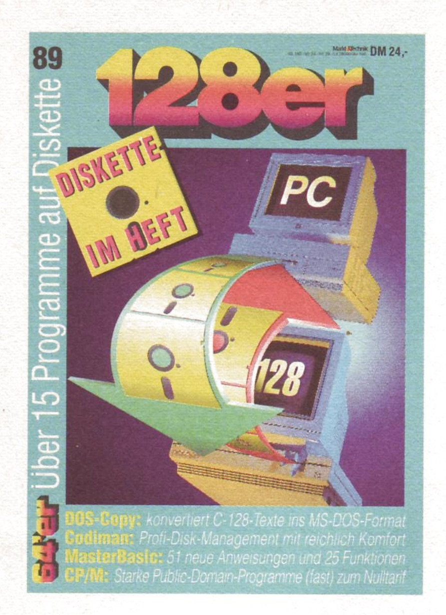
Das meistverkaufte Computerspiele- und Videospiele-Magazin

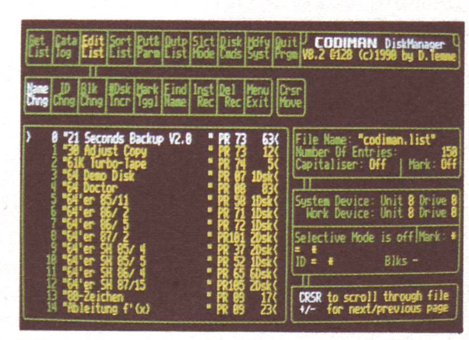
Jeden Monat ausführliche und kritische Tests von aktuellen Spielen, Informationen über kommende Highlights, Lösungen für schwer zu knackende Fälle durch "Tips&Tricks" zum Herausnehmen und Sammeln.

Das neue POWER PLAY ab sofort bei Eurem Händler!

POWER PLAY - DIE GANZE WELT DER SPIELE IN EINEM HEFT!

INHALT





Diskettenverwaltung der Extraklasse:
CODIMAN 128: erfaßt Ihre Diskothek und registriert maximal 2000 Einträge pro Datei **Seite 4**

Alle Programme aus Artikeln mit einem ■ -Symbol finden Sie auf der beiliegenden Diskette (Seite 19)

CODIMAN – Diskettenverwaltung im Griff

Turbo-Diskmanagement

Wer kennt das nicht: Dutzende von Disketten in der Diskettenbox und schon längst ist der Überblick flöten, was auf welcher Diskette ist. Sucht man ein File, hat man schon verloren: alle Disks zu überprüfen ist einfach zu langwierig. Die Lösung ist einfach: CODIMAN benutzen.

Mit CODIMAN verwalten Sie Ihre Disketten auf dem Commodore 128. Es arbeitet mit Menüsteuerung und kann zwei Laufwerke bedienen. Eine Programmdatei kann bis zu 2000 Einträge enthalten, wobei jeder Dateiname bis zu 26 Zeichen lang sein und sogar Umlaute enthalten kann.

Programmstart

CODIMAN wird im 128er-Modus mit RUN "CODIMAN" von Diskette gestartet. Das Menü der Kopfzeile offenbart alle Möglichkeiten, die CODIMAN bietet. Das zunächst leere Feld zeigt im späteren Verlauf Untermenüs sowie Felder zur Disk-Anzeige bzw. -Eingabe. Das große Ausgabefeld dient zur Bildschirmanzeige der Datei bzw. eines Disketten-Directorys. Auf der rechten Seite sind weitere vier Felder zu sehen: Das obere zeigt die Datei-Daten, darunter folgen die Systemdaten und der Modus. Unten rechts ist ein Hinweis- bzw. Rückfragefeld.

Programmbedienung

Die Anwahl der Menüpunkte erfolgt über die Cursor-Tasten (rechts bzw. links), die Auswahl über die RETURN-Taste. Alternative ist eine Anwahl durch Eingabe der unterstrichenen Buchstaben. Die Entscheidung für einen Menüpunkt führt zur Anzeige des zugehörigen Untermenüs. Hier erfolgen An- und Auswahl wie im Hauptmenü. Zurück ins Hauptmenü kommt man über MENU EXIT oder die STOP-Taste; eine direkte Anwahl eines Hauptmenüpunkts kann auch über die Eingabe des unterstrichenen Zeichens zusammen mit der SHIFT-Taste erfolgen. Eingabeanforderungen sind am blinkenden Cursor zu erkennen; die Cursor-Tasten (rechts bzw. links) sowie HOME/CLR und DEL/INST arbeiten wie gewohnt. Bei falscher Eingabe bringt die STOP-Taste den alten Text zurück, die Ausführung der Funktion wird abgebrochen. Sicherheitsabfragen werden für Ja mit RETURN oder Y (Yes) und für Nein mit STOP oder N (No) beantwortet.

Als »System Disk« wird im folgenden die Diskette bezeichnet, auf der sich CODIMAN und die Liste(n) befinden. Andere Disketten, deren Inhaltsverzeichnis beispielsweise eingelesen werden soll, heißen generell »Work Disk«.

Menüauswahl

1. GET LIST = Datei holen

Mit diesem Hauptmenüpunkt lesen Sie eine schon im CODIMAN-Format vorliegende Liste in den Speicher ein. Dort kann man sie modifizieren.

1.1 Load List

Eine CODIMAN-Datei nach Eingabe des File-Namens in den Computer laden. Eine hier evtl. vorhandene Datei wird überschrieben.

1.2 Appd List (Append List)

Eine CODIMAN-Datei nach Eingabe des File-Namens an eine bereits im Speicher befindliche Datei anhängen. Sollte kein Speicherplatz mehr vorhanden sein (zusammen mehr als 2000 Einträge), wird die Fehlermeldung »List too large« ausgegeben.

1.3 Menu Exit (STOP)

Zurück ins Hauptmenü.

2. CATALOG = Disketten-Inhaltsverzeichnis holen und bearbeiten

In diesem Modus fügt man, den Inhalt einer Diskette in die Liste ein. Mit dem Untermenüpunkt »Read Disk« werden dabei die File-Namen in eine »Diskliste« eingelesen, die nicht mit der sonst sichtbaren Liste verwechselt werden darf. Einzelne Einträge der Diskliste können dann in die richtige Liste übernommen, d.h. angehängt, werden (s. 2.2). Dabei hat man die Möglichkeit, vorher einzelne Einträge zu löschen, mehrere Files, die zusammengehören, zu einem zusammenzufassen oder auch den Namen zu wechseln. Auch die ID kann geändert werden; im Gegensatz zum Hauptmenüpunkt »EDIT LIST« werden hier aber die IDs aller Files gleichzeitig geändert, um die Identifizierung einer Diskette zu erleichtern. Mit dem Verlassen dieses Menüpunkts ist die Liste nicht verloren, so daß die Editierung jederzeit unterbrochen werden kann. Beim nächsten Aufruf wird die Liste dann erneut, so wie sie verlassen wurde, dargestellt.

2.1 Read Disk

Lesen eines Disketten-Inhaltsverzeichnisses. Die Anzeige erfolgt im Ausgabefeld. Scrollen im Catalog mit <Cursor UP/DOWN>, seitenweise mit <+/->, mit <HOME/CLR> Sprung an den Anfang bzw. an das Ende der Liste. Bei eingeschaltetem Capitaliser (s. 9.6) werden alle Anfangsbuchstaben in Großbuchstaben umgewandelt. Befinden sich bei Aufruf dieses Untermenüpunkts noch Einträge in der Diskliste, werden diese überschrieben.

2.2 Take Recd (Take Record)

Übernehmen des mit dem Cursor angewählten Eintrags in die Datei (= Löschen im Catalog und Anfügen am Ende der Datei).

2.3 Add Name

Addieren der Block-Anzahl des folgenden Eintrags zu der Block-Anzahl des angewählten. Hiermit können mehrteilige Programme unter einem Namen zusammengefaßt werden. Dabei wird der folgende Eintrag gelöscht.

2.4 Name Chng (Name Change)

Ändern des Titels des angewählten Eintrags, die Maximallänge beträgt dabei 25 Zeichen.

2.5 ID Chng (ID Change)

Ändern der Identität (ID) aller Einträge der Diskette, die Maximallänge: fünf Zeichen. Die ID dient zur Identifizierung der Disketten und muß eindeutig sein, da sonst der Sinn der Liste, nämlich das Auffinden eines Programms auf einer bestimmten Diskette, nicht erreicht wird, denn der Diskettenname wird nicht in der Datei gespeichert.

2.6 Blk Chng (Block Change)

Ändern der Block-Anzahl (max. 2999). Sinnvollerweise arbeitet man ab einer File-Größe von 664 Blöcken mit der Angabe von Diskettenseiten (s. 2.7).

2.7 #Dsk Incr (Number of Disks Increment)

Erhöhen der Anzahl der Diskettenseiten (max. 9). War der alte Eintrag in Blocks angegeben, wird »1Dsk« gesetzt. Wenn man diesen Punkt aus Versehen angewählt hat, muß man die Anzahl der Blöcke von Hand neu eingeben (s. 2.6).

2.8 Mark Tggl (Mark Toggle)

Markierung (»<« am Ende des Eintrags) ein- oder ausschalten für alle Einträge der Diskette (Bedeutung der Markierung siehe bei Punkt 7.).

2.9 Del Recd (Delete Record)

Löschen des angewählten Eintrags (verbunden mit dem Nachrücken der folgenden Einträge).

2.10 Menu Exit (STOP)

Zurück ins Hauptmenü.

3. EDIT LIST = Datei bearbeiten

In diesem Modus wird die im Speicher vorhandene Liste editiert. Diese Liste existiert unabhängig von der gerade eingelesenen Diskette des Hauptmenüpunkts »Catalog«). Die Untermenüpunkte sind aber sehr ähnlich, weshalb bei Analogien auf den jeweilig vergleichbaren verwiesen wird, wobei man aber nicht vergessen darf, daß es sich um zwei verschiedene Listen handelt, die im Hauptmenüpunkt »CATALOG« bearbeitete Diskliste dient nur zur Übernahme in die richtige, in diesem Menüpunkt editierbare.

3.1 Name Chng (Name Change)

s. unter 2.4

3.2 ID Chng (ID Change)
Ändern der ID des angewählten Eintrags (s. auch 2.5).

3.3 Blk Chng (Block Change)
s. unter 2.6

3.4 #Dsk Incr (Number of Disks Increment)
s. unter 2.7

3.5 Mark Tggl (Mark Toggle)
Ein- oder ausschalten der Markierung des angewählten Eintrags (ein »<« am Ende des Eintrags). Diese Kennzeichnung kann hilfsweise zur Selektierung benutzt werden (s. auch unter 7.)

3.6 Find Name/ID/Blocks
Finden von Titeln, IDs oder Blocks (je nach erfolgter Sortierung). Voraussetzung für die einwandfreie Funktion dieses Menüpunkts ist die Sortierung der Liste (s. auch 4.). Nach dem Aufruf dieses Menüpunkts wird mit dem Antippen eines Buchstaben ab dem ersten Namen/der ersten ID mit diesem Anfangsbuchstaben angezeigt bzw. bei »FIND Blocks« kann man mit Tasten <0 bis 9> den ersten Eintrag mit X x 100 Blocks bzw. per <D> den ersten mit »xDsk« anzeigen lassen. Durch den Druck auf CRSR DOWN/UP wird der nächste/vorhergehende Buchstabe bzw. die entsprechende Blockzahl angezeigt. Befindet man sich bei CRSR DOWN am Ende der Liste, wird wieder am Beginn der Liste angefangen. Analoges geschieht bei CRSR UP am Anfang der Liste. Beendet wird der Suchmodus mit RETURN oder STOP. Schnelleres Bewegen in der Liste kann bei längeren Listen ab 500 Einträgen sehr hilfreich sein.

3.7 Inst Recd (Insert Record)
Einfügen eines Eintrags in die Liste. Es wird ein Leereintrag erzeugt.

3.8 Del Recd (Delete Record)
Gegenstück zu 3.7: Löschen eines Eintrags, der Rest der Liste rückt nach.

3.9 Menu Exit (STOP)
Zurück ins Hauptmenü.

4. SORT LIST = Datei sortieren
Liste schnell sortieren. Dabei kommt ein besonders optimierter und angepaßter Quicksort-Algorithmus zur Anwendung, der einige Besonderheiten aufweist: Zum einen werden Groß- und Kleinbuchstaben bei der Sortierung nicht unterschieden, so daß auch bei exotischen Schreibweisen einiger Programme eine schöne Ordnung zustandekommt. Außerdem werden die Umlaute richtig einsortiert, was keineswegs selbstverständlich ist!

4.1 Name Sort
Sortieren der Liste nach Titeln.

4.2 ID Sort
Sortieren der Liste nach IDs.

4.3 Blk (Block) Sort
Sortieren der Liste nach aufsteigenden Blockzahlen.

4.4 Menu Exit (STOP)
Zurück ins Hauptmenü.

5. PUT PARM (Put List & Parameters)= Datei speichern und Parameter ändern
Unterpunkte zum Speichern der gesamten oder einer Teilliste. Außerdem kann man die Kopf- und die Fußzeile(n) des Listen-Abdrucks bestimmen.

5.1 Save All
Vollständiges Abspeichern der Datei unter einem zu bestimmenden File-Namen auf Diskette. Der Druck auf die RETURN-Taste zum Abschluß des Eingabe des File-Name ist gleichzeitig die Bestätigung, daß die richtige Diskette im Systemlaufwerk liegt. Dabei werden die Kopf- und Fußzeilen mit abgespeichert. Befindet sich eine gleichnamige Datei bereits auf der eingelegten Diskette, wird sie ohne Nachfrage ersetzt.

5.2 Save Slct (Save Selected)
Speichern der markierten Einträge auf Diskette. Mit Hilfe dieser Funktion lassen sich Fragmente aus einer einzigen großen Liste als Teilliste holen.

5.3 Head Line (Header Line)
Eingabe einer Kopfzeile. Dies geschieht in zwei Zeilen zu je 44 Zeichen, die durch ein »+« am Ende der ersten Zeile auch zu einer einzigen verknüpft werden können. Außerdem läßt sich die aktuelle Seitennummer ausgeben, indem man in die Zeile an beliebiger Stelle das Blocks-Free-Zeichen (liegt auf der Taste SHIFT-Pfund) einfügt.

5.4 Foot Line (Footer Line)
Eingabe einer Fußzeile, wieder in zwei Teilen zu je 44 Zeichen. Auch sind beide Zeilen zu verbinden, wenn man ein »+« ans Ende der ersten Zeile stellt. Die Fußzeile/n werden nur am Ende der ausgedruckten Liste gedruckt.

5.5 Menu Exit (STOP)
Zurück ins Hauptmenü.

6. OUTPUT LIST = Datei drucken
In diesem Menü kann die kreierte Liste ausgedruckt werden, wobei wieder viele Variationen möglich sind. So ist eine bis zu vierspaltige Ausgabe anwählbar. Außerdem läßt sich theoretisch jeder Drucker ansteuern, da eine äußerst komfortable Eingaberoutine für Steuersequenzen implementiert wurde, die Klartext für ASCII-Steuercodes (wie z.B. LF, HT oder ESC) sowie Strings, Dezimal- und Hexadezimalzahlen kennt. Zusätzlich kann auch jederzeit die Sekundäradresse des Druckkanals geändert werden, so daß auch die Commodore-Drucker MPS 801-803 sowie verschiedene auf dem Markt befindliche Interfaces ohne Probleme angesteuert werden können. Auch eine Centronics-Emulation am Userport wurde eingebaut (Abb.)!

6.1 Outp List (Output List)
Starten des Druckvorgangs. Er erfolgt nach den eingegebenen Parametern (s.u.). Abbruch mit STOP, Pause mit SPACE (weiter mit beliebiger Taste). Wenn der Select Mode (s. 7) eingeschaltet ist, werden ausschließlich selektierte Einträge gedruckt.

6.2 Cols Incr (Columns Increment)
Erhöhen der Zahl der nebeneinander zu druckenden Spalten (max. vier). Die jeweils gültige Spaltenanzahl ist durch die Weißfärbung des Spaltencodes erkennbar. Die benötigte Druckbreite ergibt sich wie folgt:

1 Spalte:	36 Zeichen/Zeile
2 Spalten:	75 Zeichen/Zeile
3 Spalten:	114 Zeichen/Zeile
4 Spalten:	153 Zeichen/Zeile

6.3 Tggl Hght (Toggle Height)
Umschalten zwischen voller und halber Zeichenhöhe. Die jeweils aktivierte Zeichenhöhe ist (wie beim Spaltencode) durch die Weißfärbung der entsprechenden Zeile auf dem Bildschirm erkennbar. Damit wird der Index-Modus des Druckers (falls vorhanden) genutzt, was meistens mit einer Halbierung der Druckgeschwindigkeit erkauft wird. Dabei passen nun auf eine Seite doppelt so viele Einträge, was beim Ausdruck automatisch berücksichtigt wird.

6.4 Page Form (Page Format)
Eingabe von Page Length und Paper Length (rechts neben der Untermenü-Leiste). Erlaubt sind nur Zahlen zwischen 0 und 127. Dabei gibt Paper Length die Länge der Seite in Zeilen an (für DIN A4 ist diese 72), wohingegen Page Length die Zahl der zu bedruckenden Zeilen/Seite angibt (typischer weise 66 abzüglich zweier Zeilen für die Kopfzeile ergibt 64). Ist bei Page Length eine 0 angegeben, so wird die Liste ohne Einteilung in Seiten »am Stück« gedruckt.

Menü-Übersicht

1. Get List
Datei laden oder anhängen
2. Catalog
Disketteninhalt lesen, bearbeiten und übernehmen
3. Edit List
Datei im Speicher bearbeiten
4. Sort List
Datei im Speicher sortieren
5. Put & Parm (Put List And Parameter Change)
Datei speichern/Parameter ändern
6. Outp List (Output List)
Datei auf den Drucker ausgeben
7. Slct Mode (Select Mode)
Auswahlmodus für Ausdruck oder Speichern
8. Disk Cmds (Disk Commands)
Diskettenkommandos ausführen
9. Mdfy Syst (Modify System)
Systemparameter ändern
10. Quit Prgm (Quit Program)
Programm verlassen

6.5 CSeq Mdfy (Column Sequence Modify)
Ändern der für die jeweiligen Spaltenanzahlen notwendigen Druck-Steuercodes. In der Anwendung ist das die zu wählende Schriftart (schmal, breit, fett, Elite, Pica o.ä.). Die Spaltensequenz wird beim Ausdruck (6.1) vor jeder Zeile neu ausgegeben.

6.6 HSeq Mdfy (Height Sequences Modify)
Ändern der Steuercodes für die Druckzeilenhöhe. Für die halbe Höhe (HALF HEIGHT) heißt das, daß hier die Steuerzeichen für Einschalten des Index-Modus und das Setzen des halben Zeilenabstands stehen sollten. Die Höhensequenz wird zwischen der Kopfzeile und dem Listenkopf ausgegeben.

6.7 Mode Chng (Mode Change)
Erhöhen der Drucker-Gerätenummer (4-7 oder Centronics-Treiber am Userport). Wird weiter erhöht, ändert man die Ausgabe eines LineFeeds (ASCII-Code 10) nach dem <RETURN (Code 13) auf YES oder NO und die Änderung der Codierung: entweder »Commodore«- oder »ASCII<-Code. Bei der Stellung »ASCII« werden Groß- und Kleinbuchstaben vom Commodore- in den ASCII-Code umgewandelt. Eine Sekundäradresse (auch mehrere) kann mit !zahl gesetzt werden. Die Anzeige des angewählten Druckmodus erfolgt in der Zeile unterhalb der Höhensequenzen.

6.8 Form Tggl (Format Toggle)
Umschalten des Papierformats von »SNGL« (Single) auf »CONT« (Continuous) und umgekehrt. Die augenblickliche Einstellung wird in derselben Zeile wie die Drucker-Gerätenummer (6.7) angezeigt. Wird »SNGL« eingestellt, wartet der Computer beim Ausdruck am Ende einer Seite auf den Befehl der SPACE-Taste. Vorher können Sie einen Blattwechsel durchführen. Haben Sie Endlospapier im Drucker, wählen Sie »CONT«. Dann druckt der Computer alle Seiten hintereinander. Der Vorschub zwischen den Seiten wird durch die Differenz von »Page Length« und »Paper Length« (im Bildschirm oben rechts) bestimmt (s. 6.4).

6.9 Rset Seq (Reset Sequence)
Ändern der Reset-Sequenz. Sie wird benutzt, um den Drucker vor dem Ausdruck einer Seite für die Kopfzeile bzw. am Ende einer Seite für den Vorschub und die Fußzeile am Ende der Liste zurückzusetzen.

6.10 Germ Chng (German Codes Change)
Ändern aller Codes für die Umlaute. Dabei ist nur die dezimale Eingabe (0-255) erlaubt. Zusätzlich werden die Tabellenzeichen abgefragt, wozu auch IBM-Grafikzeichen (falls der verwendete Drucker sie bietet) verwendet werden können, da sie sich hervorragend eignen. Besitzen Sie einen Commodore-Drucker älterer Generation (VC-152x oder MPS 80x/MCS 801), sind leider keine deutschen Zeichen druckbar, als Tabellenzeichen lassen sich dann die Commodore-üblichen Grafikzeichen verwendet werden. Nach der Eingabe der Zeichencodes werden die Steuersequenzen für die Umschaltung des Druckers auf den deutschen bzw. amerikanischen Zeichensatz abgefragt. Diese Sequenzen werden benutzt, wenn deutsche Umlaute und das »ß« ausgegeben werden sollen. Alle anderen Zeichen (auch die Tabellenzeichen) werden im amerikanischen Modus ausgedruckt, wodurch sich einmalige Möglichkeiten ergeben. So können die eckigen Klammern und die Tabellenzeichen gleichzeitig mit den Umlauten verwendet werden, die unter Umständen auf denselben Codes liegen.

6.11 Save Parm (Save Parameters)
Abspeichern aller eingegebenen Drucksequenzen und Einstellungen; s. auch 9.7.

6.12 Menu Exit (STOP)
Zurück ins Hauptmenü

7. SELECT MODE = Selektierungsmodus

Dieser Modus dient der Bearbeitung markierter Einträge in der Liste. Dabei können Einträge, auf die ein mit Jokern versehenes Muster paßt, markiert werden. Da man die Selektierung in den Edit-Modi (s. 2. und 3.) noch für einzelne Einträge ändern kann, erreicht man so eine sehr differenzierte Auswahl bestimmter Einträge. Die markierten Einträge können dann gelöscht (s. 7.8), ausgedruckt (s. 6.) oder abgespeichert werden (s. 5.2).

7.1 Tggl SMod (Toggle Select Mode)
Umschalten des Select Mode. Wird er eingeschaltet, sind jetzt die rechts angezeigten Selektierungskriterien aktiv. Außerdem werden die Anzahl der selektierten Einträge angezeigt sowie im EDIT-Modus diese mit einem Rückwärtspfeil gekennzeichnet. So kann man sich ein Bild von der Selektierung machen. Genau wie bei »SORT LIST« werden auch hier Groß- und Kleinbuchstaben gleich behandelt.

7.2 Name Patt (Name Pattern)
Änderung des Kriteriums »Name«. Dabei können die vom Floppy-Betrieb her bekannten Joker »*« und »?« verwendet werden. Werden auf dem Menüpunkt statt der RETURN-Taste die CRSR UP- und -DOWN-Tasten betätigt, wird die Bedingung geändert (möglich sind »=«, »= >«, »>«, »< >« (ungleich), »<«, »< =« und ohne Bedingung »-«). Die Anzahl der selektierten Einträge wird ständig aktualisiert und repräsentiert mit dem angezeigten Wert den aktuellen Zustand der Selektierung.

7.3 ID Patt (ID Pattern)
Ändern des Kriteriums »ID«. Hier gelten dieselben Bedingungen und Optionen wie bei 7.2.

7.4 Blk Patt (Block Pattern)
Ändern des Kriteriums »Blks«. Es gelten natürlich nur Blockzahlen (max. Größe 7999) bzw. Diskettenanzahlen (z.B. »5Dsk«). Joker sind nicht erlaubt. Änderung der Bedingung s. 7.2.

7.5 Mark Patt (Mark Pattern)
Ändern des Kriteriums »Mark«. Hiermit ergibt sich eine Möglichkeit, die Selektierung weiter zu verfeinern. Die Eingabe beschränkt sich auf Marke: gesetzt (»<«), gelöscht (»SPACE«) oder irrelevant (»?« oder »*«).

7.6 Clr Mark (Clear Mark)
Löschen der Marke bei selektierten Einträgen. Danach kann eine weitere Selektierung mit Setzen von 7.5 und einer Änderung der anderen Vorgaben erreicht werden.

7.7 Set Mark
Setzen der Marke bei allen selektierten Einträgen. Möglichkeiten s.o.

7.8 Reduce
Löschen aller nichtselektierten Einträge. Option ist nur nutzbringend zu verwenden, wenn 7.6 oder 7.7 nicht zufriedenstellend sind (komplizierte Selektierungen) oder zur Erstellung neuer (Teil-)Listen. Mit Vorsicht zu verwenden, gelöschte Einträge unwiederbringlich verloren!

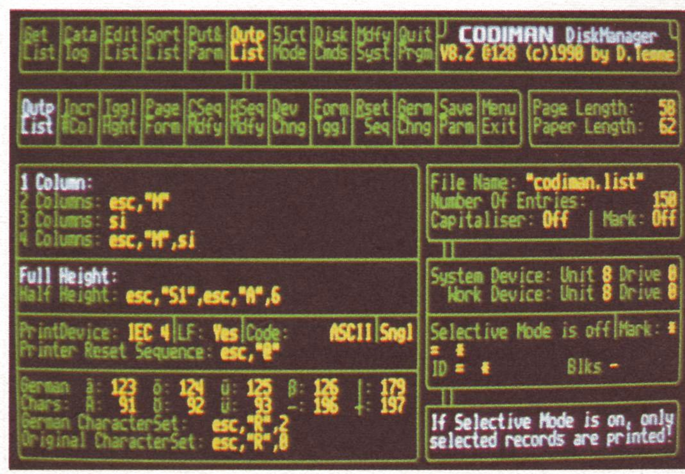
7.9 Menu Exit (STOP)
Zurück ins Hauptmenü.

8. DISK COMMANDS = Diskettenkommandos

In diesem Menü können Arbeiten ausgeführt werden, die nicht direkt im Zusammenhang mit der Liste stehen. Braucht man z.B. eine neue Diskette, so kann man sich hier eine formatieren. Auch die Ausgabe des Inhaltsverzeichnisses einer Diskette ist möglich, wenn man eine bestimmte Datei sucht. Außerdem wurde eine Änderung des Diskettennamens und der -ID implementiert.

8.1 Catalog
Ausgabe des Disketten-Inhaltsverzeichnisses. Mit der Commodore-Taste kann wie in BASIC die Ausgabe verlangsamt werden. Mit SPACE ist eine Pause möglich, wird mit weiterem Tastendruck beendet. Mit STOP wird Ausgabe auf Bildschirm abgebrochen.

8.2 Format
Formatieren einer neuen Diskette nach Angabe von Name und ID, wobei die ID fünfstellig sein kann (s. u.). Ist eine C-1571 im 1571-Modus angeschlossen, wird natürlich zweiseitig formatiert.



[2] Alle Druckereinstellungen auf einen Blick

8.3 Chng Head (Change Header)
Ändern des Kopfs eines Inhaltsverzeichnisses. Dabei kann die ID zur Identifikation fünfstellig gewählt werden. Dies wird in der Liste berücksichtigt; daher ist es sinnvoll, auch die ID fünfstellig anzulegen.

8.4 Validate
Aufräumen der eingelegten Diskette (entspricht dem Floppy-Befehl »V« und dem BASIC 7.0-Befehl COLLECT).

8.5 Menu Exit (STOP)
Zurück ins Hauptmenü.

9. MODIFY SYSTEM = System-Parameter ändern
Dieser Modus dient u.a. zur Anpassung des Programms an Ihre Rechner-Konfiguration.

9.1 Marks Off (Marks Off)
Löschen aller gesetzten Marken. Vorsicht: Sie sind unwiderrufbar verloren!

9.2 Clr File (Clear File)
Löschen des gesamten Files im Speicher und Rücksetzen der Selektierungskriterien nach einer Sicherheitsabfrage. Vorsicht: Datei ist unwiederbringlich gelöscht!

9.3 Syst Dev (System Device)
Ändern der Geräte- und Drivenummer der System-Floppy bei jedem Anwählen des Menüpunkts (erst in Gerätenummer 8, Drive 1, dann 9,0 und 9,1; ständig sichtbar im Systemdatenfenster). Die von Commodore angebotenen Einzelfloppies kennen natürlich nur Drive 0, Drive 1 ist den alten Floppies CBM 3040-8250 vorbehalten. Auf die System-Device wird bei 9.7 und beim Laden und Speichern der Datei zugegriffen. Besitzt man zwei Laufwerke, so kann man die Diskette mit CODIMAN und der/den Datei/en immer im bestimmten System-Laufwerk belassen werden.

9.4 Work Dev (Work Device)
Ändern der Geräte- und Drivenummer des Arbeitslaufwerks. Angesprochen wird es beim Einlesen neuer Einträge (2.1) und bei Diskettenbefehlen (8.). Ausführungen s. 9.3

9.5 Tggl Keyb (Toggle Keyboard)
Umschalten der Tastatur zwischen amerikanischer und deutscher Belegung. Die aktuelle Belegung wird ständig beim Aufruf des Untermenüs angezeigt. Bemerkenswert ist, daß bei deutscher Tastatur die eckigen Klammern nicht verlorengehen, da die Umlaute andere Codes haben. Beim Druck (s. 6.) wird dies auch berücksichtigt. Die deutsche Belegung entspricht bewußt nicht der von Commodore vorgegebenen, da sie unpraktisch erscheint und sich nicht für das Programm eignete, weil sie beim Umschalten in den amerikanischen Modus Verwirrung gestiftet hätte. So ist die deutsche Belegung ein Kompromiß aus amerikanischer und deutscher Tastatur, an den man sich aber schnell gewöhnt.

9.6 Lett Chng (Letter Change)
Ein- bzw. Ausschalten des »Capitalisers« (im Bildschirm rechts). Ist der Capitaliser eingeschaltet, werden beim Lesen des Inhaltsverzeichnisses im Untermenü »CATALOG« (s. 2.1) alle Anfangsbuchstaben großgeschrieben. Zu den Anfangsbuchstaben zählen auch alle Buchstaben, die nach einem Sonderzeichen (ASCII-Code 32 bis 63) stehen, außer hinter dem Apostroph. Außerdem werden die IDs bei eingeschaltetem Capitaliser vollständig in Großbuchstaben gewandelt. So werden meistens keine Änderungen am Eintrag vorzunehmen sein, außer z.B. bei römischen Ziffern.

9.7 Parm Save (Parameter Save)
Abspeichern aller Parameter des Programms. Im einzelnen sind dies: Tastatur-Modus (deutsch oder amerikanisch), zuletzt angegebener File-Name, System- und Arbeitslaufwerk sowie alle Parameter und Eingaben im »OUTPUT LIST«-Menü, von dem aus dieser Menüpunkt auch aufrufbar ist. Dabei ist CODIMAN unter seinem Originalnamen auf der Systemdiskette unbedingt erforderlich, da die Parameter im Programm-File selber auf der Diskette abgelegt werden. Dadurch wird kein File-Eintrag verbraucht; außerdem werden die gespeicherten Parameter sozusagen automatisch miteingeladen. Da man in den meisten Fällen nur einen Drucker besitzt, können alle Einträge und Parameter einmal optimal gesetzt und dann abgespeichert werden, wodurch sich dann jede spätere Änderung erübrigt und somit die Arbeit mit dem Programm vereinfacht.

9.8 Menu Exit (STOP)
Zurück ins Hauptmenü.

10. QUIT PROGRAM = Programm verlassen
Dieser Hauptmenüpunkt führt nach einer Sicherheitsabfrage einen Reset durch. Sollte man dennoch ins Programm zurückkehren wollen, kann man dies durch Eingabe von SYS 7181 erreichen (sogar ohne Verlust der Daten, da CODIMAN an einer Kennung merkt, ob sich bereits eine Datei im Speicher befindet).

Eingabe der Steuersequenzen

Im folgenden soll nun die Eingabe der Drucker-Steuersequenzen des Hauptmenüpunkts »OUTPUT LIST« (s. 6.) erklärt werden. Die Eingabe erfordert eine eigene Erklärung, weil sie sehr vielfältig ist. Zuerst muß gesagt werden, daß mehrere Eingaben durch Kommata getrennt werden können. Die möglichen Eingabeformate sind: Bytezahlen dezimal (ohne Vorzeichen, z.B. 65) oder in hexadezimal (mit vorangestelltem »$«, z.B. $1a). Um aber Eingaben so einfach wie möglich zu machen, können auch direkt die ASCII-Bezeichnungen eingegeben werden (z.B. LF oder ESC), die dann in die Steuercodes umgewandelt werden. Als letztes Format ist noch der in Anführungszeichen gebettete String zu nennen. Er kann auch mehr als ein Zeichen enthalten und wird entsprechend der Angabe in Device Change entweder nicht oder in ASCII-Code gewandelt (Beispiel: »S1«). So läßt sich jeder Drucksteuerbefehl zusammengesetzen. Beispiele finden Sie im Programm, das schon auf den Betrieb eines EPSON-Druckers oder Kompatiblen voreingestellt ist. Die Fehlermeldung »Error in input« kann auftreten, wenn Sie bei der Eingabe einen Fehler gemacht haben oder wenn die eingegebene Sequenz zu lang ist (Kapazität je Sequenz weniger als zwanzig Codes).

Sollten Sie ein Interface besitzen, das selbst Codewandlungen vornimmt, sind die Sekundäradressen hierfür auch einstellbar: mit dem Eingabebefehl <!> und folgender Sekundäradresse (0-15). So kann auch während der Sequenz auf eine andere Sekundäradresse umgeschaltet werden. Verwenden Sie das Ausrufezeichen ohne folgende Zahl, wird ohne Sekundäradresse weitergesandt. Damit sollte nun wirklich fast jeder Drucker steuerbar sein. Ein Commodore-Drucker älterer Generation (s.o.) muß mit Sekundäradresse 7 angesprochen werden, um Kleinbuchstaben zu erhalten.

Interessantes zum Programm

Mit der maximalen Länge der Liste von 2000 Einträgen mit je 32 Zeichen ist die ganze Bank 1 im C 128 belegt. Die Sort-Routine liegt (wie auch die Routinen zum Löschen und Einfügen eines Eintrags) zur Beschleunigung direkt in Bank 1, so daß keine zeitraubenden Bankumschaltungen beim Sortieren notwendig werden. Dazu arbeitet die Routine mit einer Tabelle, die es ermöglicht, beliebige Zeichen beliebig zu gewichten, um z.B. die Umlaute hinter den entsprechenden Vokalen einordnen zu können. Alle anderen Routinen liegen in Bank 0. Die Einträge werden übrigens ohne umständliche Feld-Verwaltung angesprochen. Die Blockzahlen sind inkl. der Markierung codiert, woraus sich die Maximalgröße der Blockanzahl auf 2999 ergibt, was aber ausreichen sollte (!). Das Programm arbeitet sowohl mit der 1541 als auch mit 1570/1571. Einzig das Formatieren, das Ändern der Kopfzeile und das Abspeichern der Parameter sind nicht für 80-Spur-Laufwerke wie die 1581 geeignet. Das Programm spricht die Floppy-Routinen nur über die Standard-Einsprungtabellen im ROM an, so daß auch Beschleuniger und IEC-Interfaces betreibbar bleiben sollten.

Beim Start von CODIMAN ist es unerheblich, ob die ASCII/DIN-Taste gedrückt ist oder nicht, da ohnehin ein eigener Zeichensatz in das RAM des VDC geladen und eine eigene Tastaturtabelle (siehe »MODIFY SYSTEM«) aktiviert wird. Auch Prozessorgeschwindigkeit und Ausgabe auf den 80-Zeichen-Bildschirm wird im Programm gesetzt. Es muß allerdings bemerkt werden, daß die Geschwindigkeit des Programms bezüglich Bildschirmausgabe speziell entwickelten Routinen zu verdanken ist, die das Attribut-RAM nur nach vollständig ausgegebenen Zeilen oder gar nicht beschreiben. Dadurch ergibt sich eine sehr hohe Ausgabegeschwindigkeit.

(pk/Dieter Temme)

Kurzinfo: CODIMAN 128

Programmart: Diskettenverwaltung
Bildschirmmodus: 80 Zeichen (VDC)
Laden: LOAD "CODIMAN",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: nur 80-Zeichen-Modus
Benötigte Blocks: 87
Programmautor: Dieter Temme

Datenmanager 128 – kompatibel & komfortabel

Lieferant für Mastertext 128

Die 1001te Adreßverwaltung für den C 128? Schon, aber: »Datenmanager 128« besticht durch kinderleichte Handhabung und Vielseitigkeit. Außerdem sind Adreßdateien, die man damit erzeugt, voll kompatibel zu Mastertext 128.

»Datenmanager 128« läuft nur im 80-Zeichen-Modus des Commodore 128. Das Programm sieht vor, 200 Datensätze mit maximal 15 Datenfeldern zu je 25 Zeichen (= 75000 Byte) zu verwalten. Merken Sie was? Die Speicherbank 1 (der Variablenspeicher des C128) bietet nach dem Einschalten nur 64256 Byte zur Datenaufnahme – die Zahl 200 ist daher nur realistisch, wenn man nicht alle Datenfelder bis aufs letzte Byte füllt. Außerdem empfehlen wir ausdrücklich, nicht mehr Datensätze einzurichten, da das Basic-Programm sonst bei der Suche relativ langsam reagiert. Wer dennoch mehr Einträge verarbeiten will (auf Kosten der Geschwindigkeit), muß die Variable »m« in Zeile 140 erhöhen (je nach Speicherbedarf der geplanten Datensätze und sparsamer Ausnutzung der 15 möglichen Datenfelder sind bei vorsichtiger Schätzung höchstens 300 bis 400 möglich). Besser ist es, mehrere Dateien anzulegen oder Sachgebiete zu unterteilen.

Dieser relativ geringe Dateiumfang ist durch die Art der sequentiellen Datenspeicherung bedingt. Dennoch – daraus resultiert eine äußerst nützliche Zusatzfunktion: Gespeicherte Datensätze lassen sich mit Mastertext 128 als Rundschreiben-Dateien verarbeiten. Allerdings muß man in der Mastertext-Maske den Joker <*> an den Dateinamen hängen, denn die Textverarbeitung erwartet das Suffix ».r« als letztes Zeichen. Deshalb wurde der Dateiname beim Speichern auf 14 Zeichen beschränkt (s. Punkt »3.Datei speichern«).

Datenmaske mit beliebigem Inhalt

Erfassen und Verwalten läßt sich alles: Adressen, Videos, Schallplatten, CDs, Bücher, Disketten, Kassetten usw. Sie müssen lediglich die passende Datenmaske für die vorgesehene Dateiart zusammenstellen.

Laden und starten Sie das Programm mit:

```
RUN "DATENMANAGER 128"
```

Es holt automatisch die Unterprogramme »U-PRG 1«, »U-PRG 2« und »ZEICHEN« in den Computer.

»U-PRG 1« ist ein Assembler-Programm, das den geänderten Zeichensatz »ZEICHENS« lädt und startet, hinter »U-PRG 2« verbirgt sich das Utility »Forminput« (64'er 4/90).

Nach dem Start fragt das Programm nach dem heutigen Datum (Eingabe-Format: tt.mm.jjjj). Unsinnige Eingaben werden abgefangen. Den Default-Wert (31.12.1993) kann man bedenkenlos überschreiben. Anschließend will Datenmanager 128 den Benutzernamen wissen. Ihre Angaben werden später beim Speichern gesichert und lassen sich nach erneutem Laden unter »Datei-Info« abrufen.

Anschließend erscheint das Hauptmenü (Abb. 1). Der Bildschirm ist zweigeteilt. Im oberen Drittel werden Datum, per eingebauter Kalenderfunktion berechneter Wochentag, Anzahl der vorhandenen Datensätze, zu bearbeitender Datensatz, Dateiname und ausgewählter Menüpunkt angezeigt. Der untere Screen ist für Menüs und daraus resultierende Bildschirmausgaben reserviert.

Das Hauptmenü bietet zehn Optionen, die man mit der entsprechenden Zahlentaste aktiviert:

<1> Menü 1

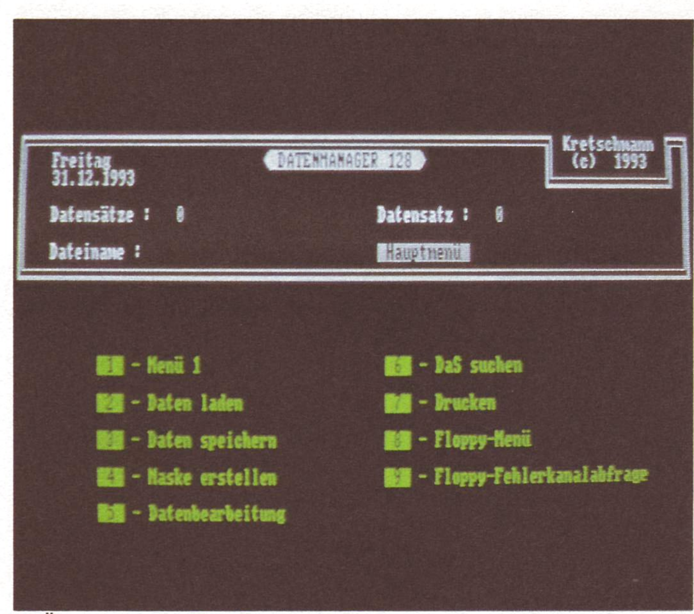
Damit ruft man fünf weitere Menüpunkte auf (es gelten ebenfalls die Zahlentasten):

<0> Datenspeicher löschen:

Diese Funktion ist wichtig, wenn man unterschiedliche Dateien nacheinander einträgt (z.B. erst Adressen, dann Videos): Auch, wenn der Datensatzzeiger auf 0 steht, könnte das Programm die vorher geltenden Daten als Vorgabe werten.

<1> Datei neu benennen:

...gibt der aktuellen Datei **im Speicher** einen neuen Namen – ohne Datenverlust oder Löschen einer alten Datei. Das Programm legt quasi eine zusätzliche Datei unter anderem Namen an. Allerdings wird sie noch nicht automatisch gespeichert!



[1] Übersichtliche Funktionen auf Tastendruck: das Hauptmenü

<2> Hauptmenü:

...zurück zum Stamm-Menü.

<3> Datei-Info:

...gibt Informationen zur geladenen Datei aus (wer war der letzte Benutzer, wann hat man die Datei zuletzt geändert, wieviel Speicher belegen die Daten usw.).

<4> Programmende:

Damit verläßt man Datenmanager 128. Befindet sich eine Datei im Speicher, erscheint eine Sicherheitsabfrage (j/n). Hat man eine Datei entworfen, die noch nicht benannt wurde, muß man den Namen eintragen. Die Daten werden dann automatisch gespeichert. Falls keine vorhanden waren, verabschiedet sich das Programm ohne Rückfrage.

<2> Daten laden

Der untere Bildschirmbereich wird gelöscht, die Frage nach dem Dateinamen erscheint. Der aktuelle Funktionshinweis in der oberen Screen-Hälfte wird jetzt invers ausgegeben. Drücken Sie nach der Eingabe <RETURN>. Denken

Sie daran, daß der Dateiname nicht länger als 14 Zeichen sein darf: Der Cursor weigert sich standhaft, mehr als 14 Byte zu akzeptieren. Die Meldung »Diskettenbetrieb. Bitte warten...« gilt als Hinweis, daß der Dateiname korrekt war und die Datei geladen wird. Probieren Sie's mit unserer Beispieldatei »ADRESSEN DEMO« aus.

Findet der Computer keine gleichnamige Datei auf der Arbeitsdiskette (oder: der Dateiname war falsch!), produziert man damit eine Fehlermeldung (Datei nicht vorhanden): Das Programm wartet auf einen Tastendruck. Dann werden Sie gefragt, ob man die Datei einrichten will: <J> generiert eine neue Datei (jedoch ohne Daten). Bei der Eingabe von <N> geht's kommentarlos zurück ins Hauptmenü.

<3> Daten speichern

Wenn vorher der Menüpunkt »Daten laden« aktiviert wurde, ist ein Dateiname vorhanden. Dann speichert das Programm die Daten automatisch mit dieser Bezeichnung – die Frage nach dem Dateinamen entfällt. Andernfalls muß man ihn natürlich angeben.

Findet das Programm weder Daten noch definierten Namen, erzeugt das Programm eine Fehlermeldung (keine Daten im Speicher). Nach Tastendruck kommt man wieder ins Hauptmenü.

ins Hauptmenü zurück oder die Maske ändern – falls Sie Ihnen doch nicht gefällt. Jetzt gilt die neue Maske als Default-Einstellung für die weitere Dateiarbeit.

<5> Datenbearbeitung

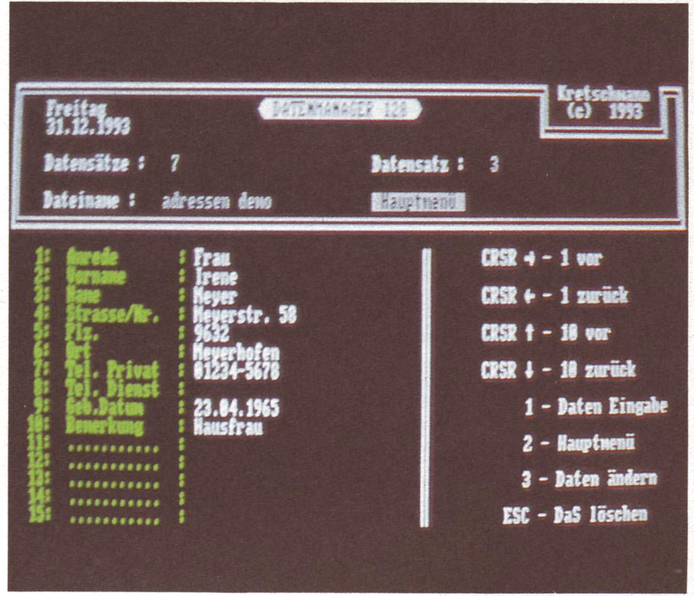
Auf der linken Bildschirmseite taucht jetzt die Eingabemaske auf. Befindet sich eine Datei im Speicher, bringt der mittlere Bildschirmteil den ersten Datensatz. Das Arbeitsmenü erscheint auf der rechten Seite (Abb. 3). Per Cursor-Tasten blättert man durch die Datei:

- <CRSR rechts>: ...holt den nächsten Datensatz,
- <CRSR links>: ...den vorhergehenden,
- <CRSR aufwärts/abwärts>: zehn Datensätze vorwärts, bzw. zehn Datensätze zurück.

Das Programm weigert sich, über den letzten Datensatz hinauszublättern. Per Tipp auf <1> erscheint ein leeres Datenblatt, der Datensatzzähler erhöht sich um 1. Jetzt kann man Daten eingeben. Bleibt ein Datenfeld ohne Eintrag, setzt das Programm automatisch den Code CHR$(141): Die Zeile wird als belegt kennzeichnet. Mit den Cursor-Tasten kann man sich durchs gesamte Datenblatt bewegen. Sobald man das letzte Datenfeld mit <RETURN> bestätigt oder es per <CRSR abwärts> verläßt, wird der Datensatz übernommen. Mit <2> kommt man wieder ins Hauptmenü.



[2] Jede Maske besteht aus individuellen Datenfeldern



[3] Beispieldatensatz einer Adreßdatei

Anmerkung: Beim Speichern generiert das Programm zwei Dateitypen: PRG und SEQ. Die SEQ-Datei enthält die eingegebenen Daten, in der PRG-Datei werden Maske und Datei-Informationen abgelegt. Der Grund: Mastertext muß für die vorgesehene Aufgabe (Rundbriefe) aufs SEQ-File zurückgreifen. Für den Anwender entstehen durch diese spezielle Datensicherung keinerlei Nachteile.

<4> Maske erstellen

Die im Programm vorhandene Maskenvorgabe erscheint auf dem Bildschirm (Adreßdatei, Abb. 2):

Das Untermenü kennt zwei Optionen:

<1> Maske erstellen:

...bietet die Möglichkeit, die Default-Maske mit individuellen Datenfeldbezeichnungen zu belegen (max. elf Zeichen). Den Vorgabetext muß man überschreiben oder mit der Tastenkombination <SHIFT CLR/HOME> löschen bzw. mit <RETURN> übernehmen. Wer den Vorgabetext löscht, aber nicht einträgt, muß sich damit abfinden, daß elf Punktzeichen im Datenfeld erscheinen. Erst nach Bestätigung aller 15 Felder wird die Maske übernommen. Jetzt kann man wieder

So ändert man Daten: Per <CRSR aufwärts/abwärts> sucht man das gewünschte Datenblatt und drückt <3>. Der blinkende Cursor erscheint wieder im ersten Datenfeld. Jetzt kann man mit den Cursor-Tasten durchs Datenblatt wandern. Änderungen werden wie bei »Eingabe« übernommen, wenns letzte Datenfeld bestätigt wurde. Die Taste <ESC> löscht jeden aktuell angezeigten Datensatz.

<6> Datensatz suchen

Damit lassen sich Datensätze nach Eingabe bestimmter Kriterien rasch ausfindig machen. <ESC> schließt die Eingabe ab, die Suche beginnt. Hier muß man nicht vorher per <RETURN> alle Datenfelder abklappern.

<7> Drucken

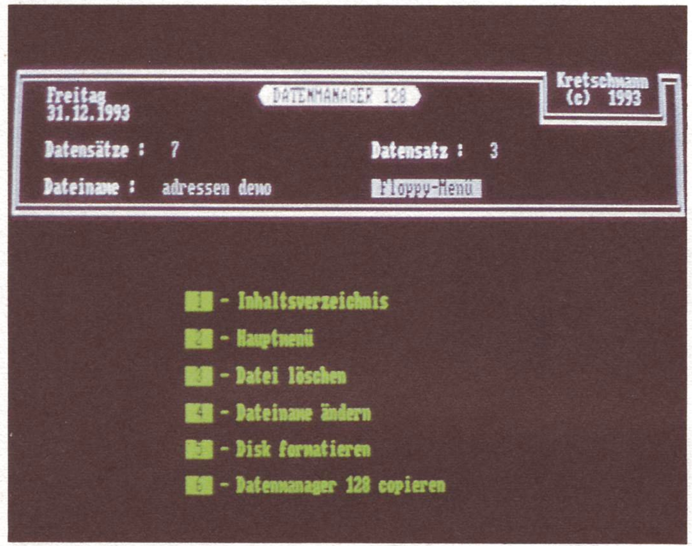
...gibt die Datensätze auf einem seriell angeschlossenen Epson-kompatiblen Drucker aus. Die Druckroutine finden Sie in den Listingzeilen ab 2350 bis 2710.

Selbstverständlich läßt sich auch nur ein bestimmter Teil der Datei drucken – zwei Datensätze sind's aber immer. Der Grund: Um Papier besser auszunutzen, setzt das Programm zwei Sätze nebeneinander. Allerdings müssen Sie dem Com-

puter die Nummern mitteilen (z.B. 2 und 3). Auch, wenn Sie zweimal dieselbe Zahl angeben, erscheinen stets zwei Datensätze nebeneinander - außer, Sie ändern das in den Listingzeilen 2520 bis 2680 (z.B. löscht man die Anweisung STEP 2 in Zeile 2520, ebenso die Sequenzen CHR$(9)+Fx$(Q+1) im nachfolgenden Programmcode.

Druckerhinweise

Achtung: Um Steuerzeichen, z.B. für die Tabellierung zum Drucker zu schicken, muß der Linearkanal des Geräts bzw. des Interface geöffnet sein. Bei der überwiegenden Anzahl Epson-kompatibler ist es die Nummer 1. Der Präsident-Drucker des Programmautors verwendet dagegen »7«, manche Interfaces arbeiten sogar mit »4«. Sehen Sie bitte im Druckerhandbuch oder in der Anleitung zum Interface nach.



[4] Diskettenmanipulationen, die das Floppy-Menü bereitstellt

Falls nötig, ändert man die Druckersekundäradresse in Zeile 2460 des Basic-Listings. Beachten Sie bei der integrierten Druckroutine, daß sie nur Großbuchstaben ausgibt (bedingt durch den Linearkanal). Kleinbuchstaben verwandelt sie aber in große, so daß dieses kleine Manko nicht weiter ins Gewicht fällt. Das ß wird unterdrückt (das gibt's nicht als Großbuchstabe, ebenso könnten manche Interfaces Schwierigkeiten mit der Codewandlung der Umlaute machen!). Sicher - per Sekundäradresse 7 läßt sich beim Drucker Klein- bzw. Großschrift aktivieren: Dann zeigt aber das Tabulator-Steuerzeichen CHR$(9) keine Wirkung mehr (die beiden Datensätze kleben zusammen!).

Das ist aber zweitrangig, denn die Hauptfunktion des Datenmanager 128 bleibt unverändert erhalten: Bei der Zusammenarbeit mit Mastertext gibt's keine Probleme - weder mit Umlauten noch Sonderzeichen. Betrachten Sie die im Programm integrierte Druckfunktion quasi als Datenkontrolleur!

<8> Floppy-Menü

Die sechs Menüpunkte (Abb. 4) ruft man ebenfalls per entsprechender Zifferntaste auf:

<1> Inhaltsverzeichnis:

...bringt das Directory aller SEQ-Dateien, die auf der Arbeitsdiskette gespeichert sind.

<2> Hauptmenü:

...zurück ins Start-Menü.

<3> Datei löschen:

...verbannt die per vorherigem Namenseintrag definierte SEQ-Datei von Diskette. Automatisch löscht die Routine auch die dazugehörige PRG-Datei.

<4> Dateinamen ändern:

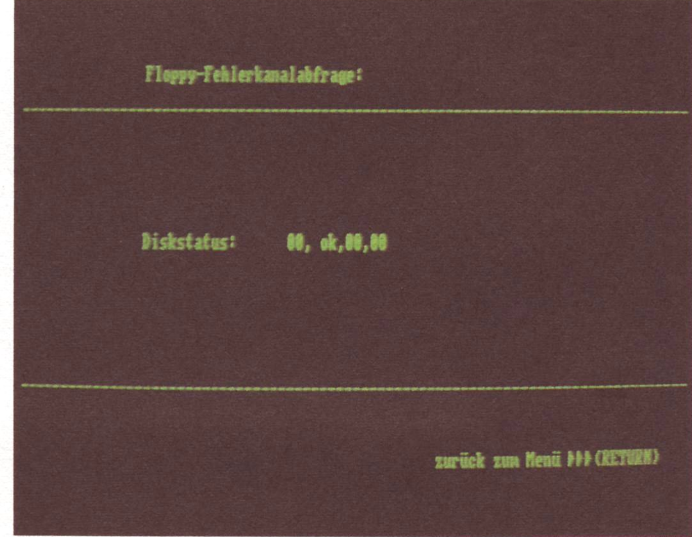
Damit lassen sich Dateien umbenennen. Es ist aber notwendig, den alten und neuen Dateinamen einzutragen. Die entsprechende PRG-Datei wird ebenfalls umgetauft.

<5> Disk formatieren:

...bereitet eine neue Diskette zur Datenaufnahme vor. Achtung: Wenn Sie den Floppytyp 1571 verwenden, wird die Diskette doppelseitig formatiert.

<6> Datenmanager 128 kopieren:

Nach dem Formatieren läßt sich Datenmanager 128 mit allen Unterprogrammen auf die neue Diskette (oder jede andere) kopieren. Beachten Sie aber, daß man alle aktuellen Daten vorher speichern muß, denn das Programm startet nach dem Kopieren automatisch von der neuen Arbeitsdisk. Alle nicht gesicherten Daten sind unwiederbringlich verloren. Am besten aktiviert man diese Funktion vor Einrichtung einer Datei.



[5] Eingabe- und Diskettenfehler fängt das Programm ab!

Bedienungsfehler haben keine Chance

Das Progamm ist mit einer Fehlerabfangroutine ausgestattet (TRAP-Funktion, wird in Listingzeile 230 initialisiert), die jedes unvorhergesehene Problem abfängt und Programmabstürze verhindert. Die Fehlermeldung erscheint im Klartext auf dem Bildschirm. Die Tastenkombination <RUN/STOP RESTORE> ist abgeschaltet. Die Solo-Taste <RUN/STOP> läßt sich zwar frei betätigen, wird aber ebenfalls per TRAP-Routine abgefangen: Damit geht kein einziges Ihrer wertvollen Daten-Bytes verloren.

<9> Floppy-Fehlerkanalabfrage

...ruft die erwähnte TRAP-Routine des Programms auf und gibt den Status der Diskettenstation auf dem Bildschirm aus (Abb. 5).

Gerade in Zusammenarbeit mit Mastertext 128 beweist Datenmanager 128 seine Stärke. Wer aber keine Rundschreiben verschicken, sondern lieber seine Briefmarken- oder Münzsammlung katalogisieren will, findet mit dieser Dateiverwaltung ebenfalls einen unentbehrlichen Helfer. (bl)

Kurzinfo: Datenmanager 128

Programmart: universelle Dateiverwaltung
Bildschirmmodus: 80 Zeichen (VDC), Farb- oder S/W-Monitor
Laden und starten: RUN "DATENMANAGER 128"
Besonderheiten: Adreßdateien sind kompatibel zu Mastertext 128 V2.0
Benötigte Blocks: 115
Programmautor: Diethelm Kretschmann

Berechnungsprogramm

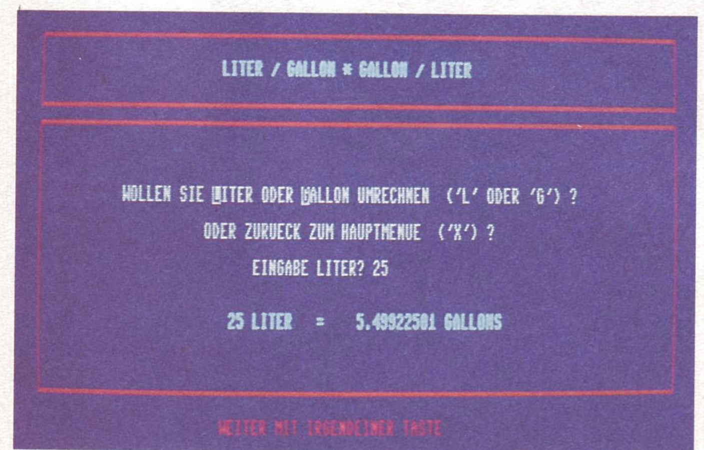
Meter sind gefragt!

Nicht immer hat man bei Arbeiten in Mathe oder Physik eine Tabelle zur Hand, die englische oder amerikanische Einheiten ins metrische System umrechnet. Der C 128 erledigt das spielend.

Nach dem Laden stehen dem Nutzer verschiedene Umrechnungen von bekannten amerikanischen und englischen Maßeinheiten ins deutsche System und umgekehrt zur Verfügung. Insgesamt wurden 17 Maßeinheiten berücksichtigt (s. Tabelle). Die Palette reicht von Kilometer und Meile, über Liter und Gallons, bis hin zu Celsius und Fahrenheit. Die Genauigkeit der Berechnungen beträgt bis sechs Stellen hinter dem Komma. Die Umrechnungen erfolgen in einzelnen Untermenüs (Abb. 2 bis 4). Dort wird gefragt in welchen Wert umgerechnet werden soll. Danach gibt man den Wert ein, den man umrechnen will. Nach Ausgabe des Ergebnisses befindet man sich weiter im Untermenü. Die Umrechnung wird mit einer Genauigkeit mit bis zu sechs Stellen hinter dem Komma vom C128 vollzogen. Auf- bzw. Abrunden bleibt dem Anwender des Programms überlassen. Mit <X> gelangt man ins Hauptmenü (Abb. 1) und mit <Q> wird das Programm, nach einer Sicherheitsabfrage, mit Reset verlassen. (L. Detering/lb)



[1] Umrechnungen von Maßeinheiten per Tastendruck

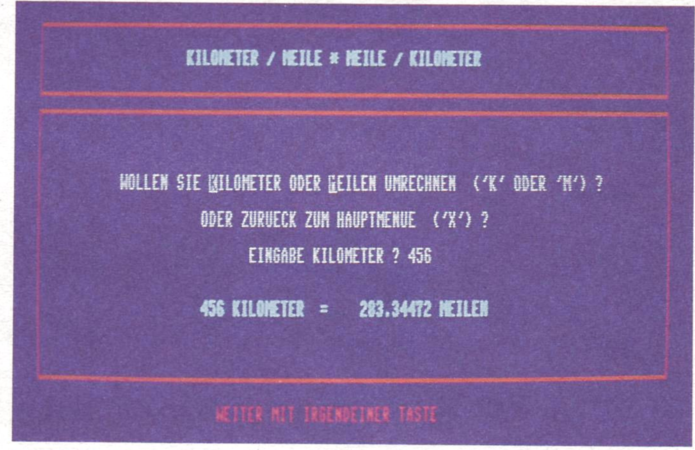


[4] Liter werden zu Gallonen

CELSIUS / FAHRENHEIT * FAHRENHEIT / CELSIUS

WOLLEN SIE CELSIUS ODER FAHRENHEIT UMRECHNEN ('C' ODER 'F') ?
ODER ZURUECK ZUM HAUPTMENUE ('X') ?
EINGABE CELSIUS? 28
28 CELSIUS = 82.4 FAHRENHEIT
WEITER MIT IRGENDEINER TASTE

[3] Grad Celsius in Fahrenheit, der exotischen Temperatureinheit



[2] Ein Untermenü – hier werden Längen umgerechnet

Die Umrechnungsmöglichkeiten (Tabelle)

	Einheiten	physik. Maß
Kilometer	Mile (Meile)	Länge
Meter	Yard	Länge
Meter	Foot (Fuß)	Länge
Zentimeter	Inch (Zoll)	Länge
Hektar	Acre	Fläche
Quadratkilometer	Square mile (Quadratmeile)	Fläche
Quadratmeter	Square yard	Fläche
Quadratmeter	Square foot	Fläche
Quadratzentimeter	Square inch	Fläche
Kubikmeter	Cubic yard	Volumen
Kubikzentimeter	Cubic inch	Volumen
Liter	Pint	Volumen
Liter	Gallon	Volumen
Tonne	Ton	Masse
Kilogramm	Pound (engl. Pfund)	Masse
Gramm	Ounce (Unze)	Masse
Celcius	Fahrenheit	Temperatur

Kurzinfo: Umrechnungen

Programmart: Rechenprogramm
Bildschirmmodus: 80 Zeichen (VDC)
Laden: DLOAD"UMRECHNUNGEN"
Starten: nach dem Laden <RUN> eingeben
Besonderheiten: Umrechnung vom metrischen System ins amerikanische oder englische und umgekehrt
Benötigte Blocks: 119
Programmautor: Lothar Detering



Wow! Soviel
Top Themen p
64'er
Sonderhefte
Jetzt zum Nachbestellen
zum Preis von DM 16.- je Heft,
128er Sonderhefte für DM 24,-.
Ordnung im Archiv mit der praktischen Sammelbox
für nur DM 14,-.
Einsteiger
SH 50: Starthilfe
Alles für den leichten Einstieg/ Heiße Rythmen mit dem C64/ Fantastisches Malprogramm
SH 62: Erste Schritte
Exbasic Level II: Die Super-Basic-Erweiterung RAM-Exos: Floppy-Speeder der Extraklasse
SH 74: Einsteiger
Basic 3.5.: Basic-Erweiterung mit Hires-Grafik-Befehlen / FOBS V1.0: Floppy Operationen per Benutzeroberfläche
Anwendungen
SH 68:
Faszination Sternenhimmel. Der Blick in den Kosmos/ Kreuzworträtsel zum Selbermachen
SH 78:
Grafic-Calc:Grafische Auswertung von Jahresbilanzen/MAS V1.0: Übersichtliche Schulnotenverwaltung
SH 81:
Paint Mania: Zeichenprogramm der Superlative/ Maestro: Eigenen Sound auf Knopfdruck komponieren/ Disk Tools V6.5: Enttarnt jedes Byte auf Diskette
SH 86:
Database 2.0: Universelle Datenbank mit starken Rechenfunktionen/ Stamp Collection Kit: Archiv für Ihre Briefmarken/ außerdem: Autokauf, Haushaltskasse & Girokonto
GEOS
SH 48: GEOS-Erweiterungen
Geotext - neuer, schneller Texteditor für Geowrite/Workshop zu Geopublish
SH 59: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks
SH 80: GEOS
Lottoblock: Statistische Gewinnzahlauswertung mit Tippvorschlägen/ Finanzen: Welche Geldanlage ist die Beste ?
Tips, Tricks & Tools
SH 65:
Streifzug durch die Zeropage/ Drucker-Basic: 58 neue Befehle zur Printer-Steuerung/ Multicolorgrafiken konvertieren/ über 60 heiße Tips&Tricks
SH 77:
Amica-Konvert: 6 Malprogramme tauschen Grafik aus/ Disc-Basic: Floppybefehle kurz und prägnant prüfen
Hardware
SH 67:
Wetterstation.Temperatur, Luftfeuchtigkeit und -druck-messen/ DCF Funkuhr und Echtzeituhr/ Daten konvertieren: vom C64 zum Amiga, Atari ST und PC
SH 83: Floppy
4 Kopiertools für komplette Backups und Einzel-Files/ Spur 36 bis 40: 95 Blocks zusätzlich auf Diskette viele Tips&Tricks
SH 84: Hardware
C64 kompakt: Computer und Floppy in einem Gehäuse (Umbauanleitung)/ Midi-Interface: C64 mit Keyboard/ Tiny-EPROMer: EPROM-Brenner im Selbstbau unter 30 DM
DISKETTE IM HEFT
64'er Spiele
11 Superspiele für Drahtseilnerven
Daten-Agent 00X: Jagd im All
Zoff im Geisterschloß
Charakter-Editoren: Spielelösungen, Pokes und Cheats
Alle Programme auf Diskette
DISKETTE IM HEFT
64'er HARDWARE
Floppy-Speeder für 10 Mark
8-Kanal-Schaltinterface
Sieben Sinne für den C 64
Der Edel-C 64 mit Floppy
10 Bauprojekte mit Programmen
DISKETTE IM HEFT
Alle Hefte mit randvoller Programm-Diskette
Top Spiele mit Diskette

Know how!
plus Diskette!

Programmier-Sprachen

SH 71: Assembler
Kursus/ Komplettpaket/ Befehlsposter/ Tips & Tricks/ Leserfragen

Grafik

SH 55:
Amica Paint: Fantastisches Malprogramm für Hobby-Grafiker, mit allen Up-Dates

SH 75:
Interlace 64: 136 Farben und 640x200-Pixelgrafik und 80-Zeichen-Bildschirm/ 3D-Animation mit Hi-Eddi

Drucker

SH 72:
Publish 64: DTP-Einstieg/ Topprint: Druckt Briefköpfe, Schilder und Grußkarten/ Test: Drucker unter 1.000 DM

C 128

SH 58:
Übersichtliche Buchhaltung zuhause/Professionelle Diagramme

SH 64: Anwendungen:
Amerikanisches Journal/ Grundlagen: CP/ M, das dritte Betriebs-system/ VDC-Grafik: Vorhang auf für hohe Auflösung

SH 76:
Music Master: Professionelle Datenbank zur LP- und CD-Sammlung/ Prüfungsaufgaben: Idealer Helfer vor jedem Examen

SH 82:
Mini-Micro: Kopiert 1571-Disketten zur 1581/ Pro-Book 128: Komfortable Datenbank für den Bücherwurm

Spiele

SH 2: Top Spiele
10 Game-Trainer und 2 Longplays/ Katakis-System: Spiele programmieren wie die Profis/ Tolle Tips für höhere Level und Unsterblichkeit

SH 3: Top Spiele
Action Jump'n Run Logik/ Tips, Tricks & Tools

SH 54:
15 tolle Spiele auf Diskette/ Der Sieger unseres Programmierwettbewerbs: Crillion II/ Ein Cracker packt aus: Das ewige Leben bei kommerziellen Spielen

SH 60: Adventures
8 spannende Abenteuerspiele/ 2 Komplettlösungen und viele Tips, Adventures selber programmieren

SH 61:
20 heiße Super-Games für Joystick-Akrobaten/ Cheat-Modi und Trainer POKEs zu über 20 Profi-Spielen/ Krieg der Kerne: Grundlagen Spieleprogrammierung

SH 66:
15 Top-Spiele mit Action und Strategie/ Mondlandung: Verblüffend echte Simulation und Super-Grafik/ High-Score-Knacker: Tips & Tricks zu Action-Games

SH 73:
Action bis Adventure: 10 Spiele zum Kampf gegen Fabelwesen/ Preview/ Tips & Tricks/ Kurse/ Game Basic/ Mission II/ W.P.Tennis II/ Omnibus GmbH/ Mic's Push'em

SH 79:
25 starke Mega-Games/ Exis: Gefährlicher Satelliten-schleppdienst/ Tips & Spielelösungen

SH 85:
11 Super-Games für stählerne Nerven/ Datenagent 00X: Noch 12 Stunden bis zum Weltuntergang/ Kick'n kill: Irrwitzigees Jump-and-Run-Spiel für Joystick-Akrobaten

Nur noch hier erhältlich!

Jetzt sofort Bestellen - per Post oder FAX !

64'er

Ordnung im eigenen Archiv für DM 14.-

Hier war leider jemand schneller, doch null Problem: Einfach schreiben und bestellen bei 64er Leserservice CSJ, Postfach 140 220, 8000 München 5, Telefon 089/ 240 132 22, FAX: 089/ 240 132 15

BESTELLCOUPON

Ich bestelle folgende 64er Sonderhefte:

	SH-Nr.	SH-Nr.	SH-Nr.	
____Sonderhefte mit Diskette je 16,- DM				____ DM
____Sonderhefte"128er" je 24.-DM				____ DM

Ich bestelle______Sammelbox(en) ____ DM
zum Preis von je DM 14,- **Gesamtbetrag** ____ DM

Ich bezahle den Gesamtbetrag zzg. Versandkosten nach Erhalt der Rechnung.

Vorname,Name

Straße, Hausnummer

PLZ, Wohnort

Datum / Unterschrift

Schicken Sie bitte den ausgefüllten Bestellcoupon an: 64er Leserservice, CSJ, Postfach 140 220, 8000 München 5, Telefon: 089/ 240 132 22 oder bequem per Telefax: 089/ 240 132 15

Zenec V2.0 – Druckertool

Phantastische Schriften mit dem NEC-P 2200

Für viele 9-Nadel-Drucker gibt es bereits Zeicheneditoren, die erstaunliche Schriftqualität beim Ausdruck beweisen. Aber warum nicht auch die Qualität eines 24-Nadel-Druckers wie des NEC-P 2200 nutzen? Genau das erreicht unser Programm »Zenec V2.0«. Damit wird es zur Spielerei, eigene Zeichen zu generieren und mit dem Drucker auszugeben.

Die Möglichkeiten sind faszinierend: Ein Editierfeld von nicht weniger als 24 Zeichen und 37 Spalten reicht aus, um eine Vielzahl unterschiedlicher Zeichen zu kreieren. Sie können Zeichensätze vier verschiedener Matrix-Typen editieren, auf Diskette speichern, von Diskette laden und natürlich auch zum Drucker schicken. Kleine Hilfen wie Bewegen der Zeichen in der Matrix, Wenden und Kopieren auf andere Zeichen sind für ein Programm dieser Güte selbstverständlich.

Die neuen Zeichensätze können problemlos zum Drucker geschickt werden, wo sie anschließend für die Anwendung unter einer Textverarbeitung (z.B. Mastertext 128) zur Verfügung stehen.

Als Hardware-Voraussetzung benötigen Sie den C128 im 80-Zeichen-Modus und einen NEC-P 2200, den Sie mit einem geeigneten Kabel am Userport des Computers anschließen. Und natürlich benötigen Sie nach dem Start des Programms etwas Kreativität, um Schriftzeichen zu editieren.

So arbeiten Sie mit dem Programm

Laden und starten Sie mit

```
RUN "ZENEC V2.0"
```

Achtung: Die ASCII/DIN-Taste darf beim Start nicht gedrückt sein. Wenn doch, fordert das Programm Sie vorher auf, die Taste zu lösen.

Im ersten Eingangsmenü (Abb. 1) entscheiden Sie zunächst, welche der vier Matrix-Typen Sie bearbeiten wollen. Diese wählen Sie über die Tasten <A> bis <D> aus. Anschließend stehen die verschiedenen Funktionen des Hauptmenüs zur Verfügung:

<F1> Edit

Mit der Funktionstaste <F1> gelangen Sie in den Zeichengenerator (Abb. 2), dessen Funktionen noch ausführlicher erläutert werden.

<F3> Load

Mit der Funktionstaste <F3> laden Sie einen bereits fertigen Zeichensatz von Diskette. Dabei können Sie diese komfortabel aus dem Directory heraus laden. Das Directory wird übersichtlich auf dem Bildschirm dargestellt. Mit den Cursor-Tasten wird es auf- und abgescrollt. Der Dateiname unter dem reversen Balken wird durch Drücken mit der RETURN-Taste geladen.

Ein Dateiname für einen Zeichensatz muß nicht mit einem Punkt und dem Schrifttyp enden. Ist das 16. Zeichen ein A, B, C oder D, versteht es das Programm als Schrifttyp. Fehlt der Schrifttyp, wird er vom Programm auf D gesetzt.

Sollten Sie diese Funktion unbeabsichtigt angewählt ha-

```
ZS: DEFAULT          .? *              * ZENEC V2.0 fuer C128 und NEC-P 2200

                Schrift-Typen

                Typ            B max.          A+B+C

                A - PS         37              beliebig
                B - LQ10       29              36
                C - LQ12       23              30
                D - Draft       9              12

                Bitte A,B,C oder D waehlen

F1 = Edit     F3 = Load     F5 =Save     F7 = Druck     T = ZS-Typ aendern
(c) 12.07.1988  by Hannes Mueller, Birchweg 3, CH-8200 Schaffhausen, Schweiz
```

[1] Im Eingangsmenü von »Zenec V2.0« bestimmen Sie den Schrifttyp, den Sie editieren

ben, bringt die ESC-Taste Sie wieder ins Hauptmenü. Zwei fertige Zeichensätze finden Sie auf der Sonderheftdiskette.

<F5> Save

Gespeichert wird ein editierter Zeichensatz über die Funktionstaste <F5>. Das Programm schlägt einen Filenamen (Default) vor, den Sie durch ein Return übernehmen können. Bei einer Neueingabe werden maximal 14 Zeichen angenommen, da das Programm automatisch einen Punkt und den im Eingangsmenü gewählten Schrifttyp-Buchstaben an den Dateinamen anhängt.

Achten Sie darauf, jeweils nach Änderung eines Zeichensatzes einen neuen Dateinamen zu vergeben, da bei der Übernahme des alten Namens der alte Zeichensatz nicht überschrieben wird. Das Programm gibt dann eine Fehlermeldung aus. Lassen Sie sich von der flackernden LED des Laufwerks nicht irritieren: nach dem nächsten erfolgreichen Zugriff auf Diskette ist wieder alles in Ordnung.

<F7> Druck

Wenn Sie die Funktionstaste <F7> drücken, wird der aktuelle Zeichensatz zum Drucker übertragen. Dabei erfolgt gleichzeitig ein Probedruck des Zeichensatzes mit Angabe des ausgewählten Dateinamens.

Mit dieser Funktion wird auch ein Zeichensatz in den Drucker für die Anwendung unter einer Textverarbeitung geladen.

<T> Zeichensatz-Typ ändern

Haben Sie im Eingangsmenü bei der Auswahl des Schrifttyps eine falsche Eingabe gemacht, kehren Sie über die Taste <T> zu diesem Menü zurück und wählen Sie dann den Schrifttyp, den Sie wirklich editieren wollen.

Ein komfortabler Editor

Sobald Sie vom Hauptmenü aus über die Funktionstaste <F1> den Editor angewählt haben, sehen Sie auf dem Bildschirm das Editierfeld. Wenn Sie vorher keinen fertigen Zeichensatz von Diskette geladen haben, ist das Feld leer (Abb. 2). Ansonsten erscheint ein Zeichen aus einem vorher geladenen Zeichensatz und Sie können sich einen Eindruck von dem Zeichensatz verschaffen (Abb. 3).

Ausgezeichnete Programmierung des VDC erlaubt eine Bildschirmdarstellung von insgesamt 33 Zeilen bei 80 Spalten. Dieser Platz reicht aus, um alle Funktionen am Bildschirm sehr übersichtlich auszugeben.

Sehr hilfreich ist z. B. die Anzeige der dezimalen Werte der einzelnen Bytes. Am unteren Rand erkennen Sie die Tastenbelegung für die Bearbeitung der Zeichen.

Der jeweils aktive Modus wird am oberen Bildschirmrand blinkend dargestellt.

Die Funktionen des Editors im Überblick:

Cursor-Tasten

Mit den Cursor-Tasten bewegen Sie sich wie gewohnt innerhalb des Editierfeldes.

<HOME>

Auch die HOME-Taste schickt den Cursor wie üblich an die erste Position des Feldes.

<CLR>

Die Taste löscht das gerade bearbeitete Zeichen.

<+/-> Punkt ein/aus

Mit diesen beiden Tasten setzen oder löschen Sie einen Punkt. Der Cursor bewegt sich dabei in der aktuellen Cursor-Richtung weiter. Dies ist zunächst sicher etwas gewöhnungsbedürftig, wenn Richtungsänderungen vorgenommen werden sollen. Hat man sich aber darauf eingestellt, wird Editieren so recht einfach.

<F1> und <F3>

Diese beiden Funktionstasten lassen Sie um ein Zeichen vorwärts oder rückwärts springen. Im Feld ASCII sehen Sie dabei jeweils das ausgewählte Zeichen mit seinem dezimalen ASCII-Wert angezeigt.

<F5> Parameter setzen

Wählen Sie diesen Menüpunkt, springt der Cursor in die Anzeige der drei Parameter, die für linken Leerraum (Left), Zeichenkörper (Body) und rechten Leerraum (Right) stehen.

Mit den Cursor-Tasten auf/ab werden die Werte verändert und mit RETURN bestätigt. Die Werte für die drei Parameter müssen den Schrifttypen entsprechen. Stehen die Werte auf Null, nutzt das Programm das entsprechende Zeichen aus dem Drucker-ROM.

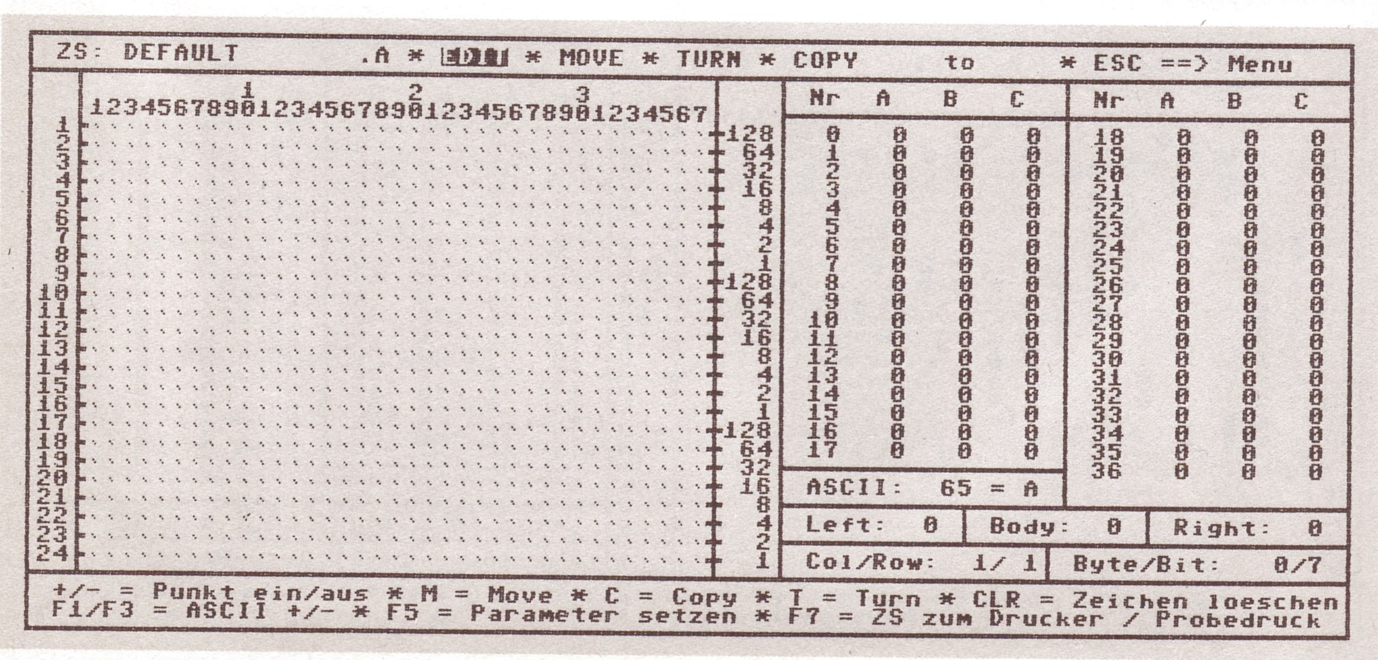
Unsinnige Werte werden vom Programm automatisch für den Ausdruck zurechtgestutzt.

<F7> Zeichensatz zum Drucker/Probedruck

Der Zeichensatz wird zum Drucker geschickt, der sofort einen Probedruck beginnt. Dies ist die gleiche Funktion wie im Hauptmenü.

<M> Move

Über die Cursor-Tasten wird ein bearbeitetes Zeichen in



[2] Der Editier-Bildschirm ist dank der Darstellung mit 33 Zeilen sehr übersichtlich

diesem Modus in alle vier Richtungen bewegt. Das Scrollen durch das ganze Editierfeld erlaubt vielfältige Variationen eines Zeichens. Mit der ESC-Taste wird dieser Modus wieder verlassen.

<C> Copy

Ein bearbeitetes Zeichen kann in diesem Modus an die Stelle eines anderen Zeichens kopiert werden. Wählen Sie dazu das Zielzeichen aus mit <CRSR-aufwärts/abwärts>. Mit RETURN wird der Kopiervorgang ausgeführt.

<T> Turn

Mit den Cursor-Tasten wird ein bearbeitetes Zeichen um die horizontale oder vertikale Achse gespiegelt. Zum Editieren gelangen Sie mit <ESC> zurück.

<ESC>

Aus dem Editiermodus springen Sie mit dieser Taste wieder ins Hauptmenü.

Die Fähigkeiten des Druckers nutzen

Der parallele 24-Nadel-Drucker NEC-P 2200 erlaubt es, User-definierte Zeichensätze zu verwenden. Der zugelassene Definitionsbereich für die neuen Zeichen geht dabei von ASCII 0 bis 127. Werden jedoch Werte unter dezimal 33 und oberhalb 126 verwendet, können eventuell bestimmte Control-Funktionen, die von den neuen Zeichen überschrieben werden, nicht mehr genutzt werden.

Die Zeichenmatrix umfaßt 37 Spalten und 24 Zeichen. Die Anordnung der Daten selbstdefinierter Zeichen ist zwingend vorgeschrieben.

Jeder der Spalten werden von oben nach unten 3 Byte zugeordnet. Dabei entspricht ein gesetztes Bit einem gesetzten Punkt. Das Zeichen selbst wird in drei Teile gegliedert: A = linker Leerraum, B = Zeichenkörper und C = rechter Leerraum. Diese drei Parameter werden in Anzahl von Spalten angegeben. Für die verschiedenen Druckarten sind feste Größen vorgeschrieben, die Sie im Eingangsmenü als Vorgaben wiederfinden.

Tips für die Anwendung

Natürlich wollen Sie die editierten Zeichensätze innerhalb einer Textverarbeitung oder anderen Programmen, die auf einen schönen Zeichensatz angewiesen sind, anwenden. Starten Sie dazu zunächst »Zenec V2.0« mit RUN von der Diskette und laden den entsprechenden Zeichensatz, den Sie verwenden wollen.

Nachdem Sie den Drucker eingeschaltet haben, wird der Zeichensatz zum Drucker geschickt. Nun wird der Computer aus- und anschließend wieder eingeschaltet. Achten Sie darauf, daß der Drucker den Computer-Reset nicht mitmacht, da sonst alle Daten wieder verschwunden sind.

Der Drucker muß selbstverständlich aktiv bleiben, da er sonst wieder die eingebauten Zeichen aus dem Drucker-ROM verwendet.

Nun wird eine Textverarbeitung geladen, innerhalb der Sie mit Hilfe der entsprechenden Steuercodes zwischen dem Originalzeichensatz und Ihrem eigenen Zeichensatz umschalten können.

Achten Sie bei der Textverarbeitung darauf, daß Sie einen vorhandenen Steuercode für einen Reset aus der Steuerzeichentabelle entfernen. Dies ist z. B. bei »Mastertext 64« und bei »Mastertext 128« notwendig, da durch den Reset des Druckers der eigene Zeichensatz verlorengeht.

Bei »Vizawrite« für den C128 ist die Einstellung des Druckers auf dem parallelen Anschluß vorzunehmen. Zusätzlich sollten Sie sowohl Linefeed als auch Formfeed einschalten.

»Protext 128« erlaubt die Verwendung des eigenen Zeichensatzes über den Druckertreiber »parallel-1«. Für einen guten Ausdruck ist auch hier ein zusätzliches Linefeed erforderlich.

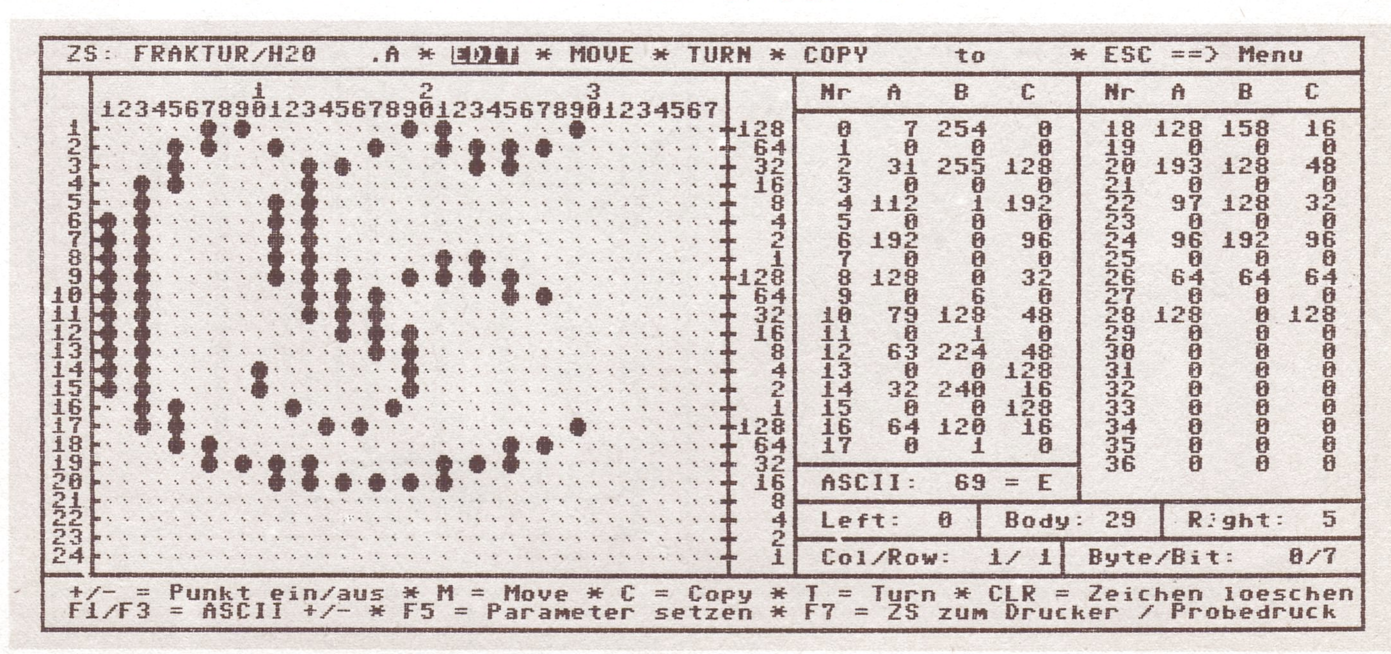
Da »Zenec V2.0« nach dem Übertragen des Zeichensatzes sofort mit einem Probeausdruck beginnt, kann das erste Blatt nicht mehr verwendet werden. Sie sollten es aber für einen ersten Testdruck benutzen. Dies ist sinnvoll, da manche Textverarbeitungsprogramme vor dem ersten Text einige Grafikzeichen ausdrucken. Sollten Sie das auch bei Ihrer Textverarbeitung feststellen, braucht Sie das nicht zu irritieren. Nach dem Druck dieser Grafikzeichen stehen alle Zeichen Ihres Zeichensatzes für den Ausdruck zur Verfügung.

Das Programm ist vollständig in Assembler geschrieben.

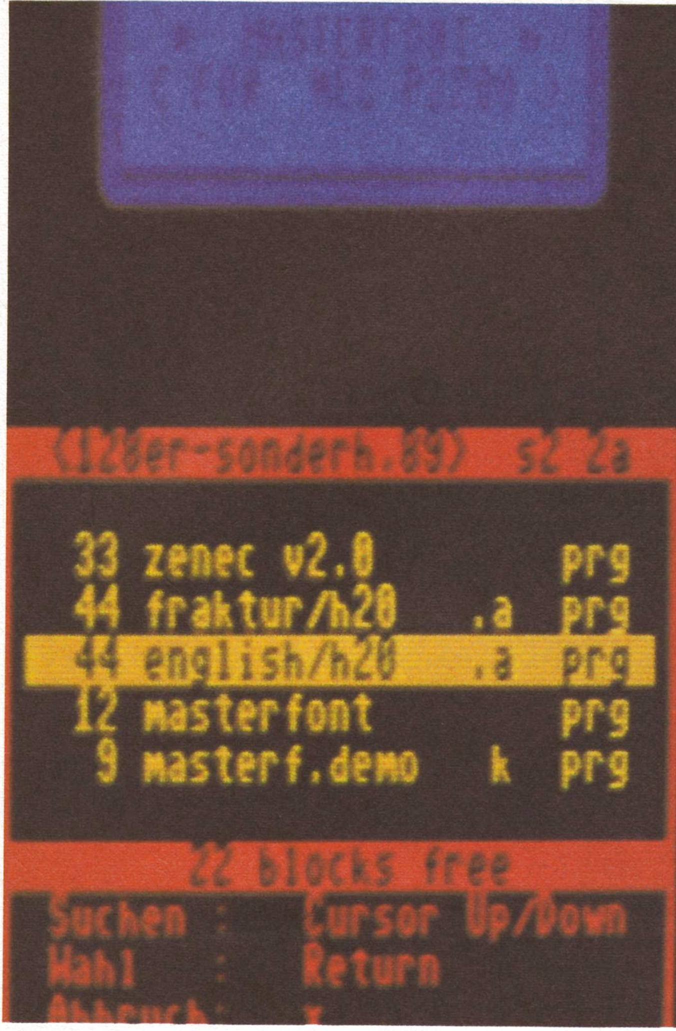
Es belegt dabei in der Bank 0 den Bereich von $1c01 bis $3be9 und ist mit einer Basic-Startzeile versehen.

Der Zeichenspeicher liegt ebenfalls in Bank 0, und zwar von $4000 bis $6ac1. Die beiden Basic-ROMs sind ausgeblendet. Der 40-Zeichen-Bildschirm dient als Hilfsspeicher.

Für die Darstellung der Menüs wird durch intensive Nutzung des Prozessors der benötigte Speicherplatz möglichst gering gehalten.



[3] So könnte ein neues Zeichen aussehen. Editiert ist hier der Buchstabe <E>.



[4] Masterfont transportiert die Zenec-Zeichensätze komfortabel zum Drucker

Kurze Beschreibung des Programmaufbaus

Spalten und 24 Zeichen erfordert für jedes Zeichen 111 Byte zu acht Punkten. Hinzu kommen die drei Steuerbyte A, B und C. Für den Zeichensatz werden also 96 mal 114 Byte benötigt, die hintereinander im Speicher abgelegt sind und über eine Anfangsadreßtabelle mit einem Zeropage-Zeiger angesprochen werden.

Das Directory wird beim Ladevorgang erst in den Speicher ab $8000 eingelesen und dort aufbereitet. Beim Laden wird der Schrifttyp interpretiert. Die eigentlichen Load- und Save-Routinen sind standardisiert.

Da die Steuerbefehle für 24-Nadel-Drucker weitgehend standardisiert sind, dürfte »Zenec V2.0« auch mit anderen 24-Nadlern zusammenarbeiten.

Masterfont: Die Erweiterung

Masterfont von unserem Autor Peter Hülstede ist eine Erweiterung, die Zenec V2.0 noch um einiges verbessert. Wir drucken es deshalb hier mit ab (Abb. 4).

Die mit Zenec erstellten Zeichensätze können z.B. beim Arbeiten mit einem Textverarbeitungsprogramm verwendet werden. Ein Nachteil ist dabei allerdings, daß man in einem Ausdruck nur einen Zeichensatz einsetzen kann, da man Zenec braucht, um den Zeichensatz zum Drucker zu übertragen. Außerdem erzeugt Zenec nach der Übertragung jedesmal einen Probeausdruck in zwei Schriftgrößen. Masterfont ist ein Übertragungsprogramm für die mit Zenec entworfenen Zeichensätze, mit dem man diese Probleme umgehen kann.

Die Programmbedienung

Nach dem Start von Masterfont wird man aufgefordert, eine Diskette mit Zenec-Zeichensätzen einzulegen und eine Taste zu drücken. Daraufhin wird das Directory der eingelegten Diskette geladen und man kann sich den zu ladenden Zeichensatz mit der Cursor-Taste aus dem Directory aussuchen. Nach einem Druck auf die Return-Taste wird der gewählte Zeichensatz geladen. Wenn man "x" drückt, läßt sich eine neue Diskette einlegen. Nach dem Laden erscheint ein Menü auf dem Bildschirm, in dem man mit den Tasten "1" bis "3" zwischen den drei Ausgabemöglichkeiten wählen kann. Außerdem kann man mit den Tasten "n" und "r" zwischen normaler und reverser Ausgabe umschalten. Menüpunkt 1 überträgt den Zeichensatz zum Drucker, der an den User-Port angeschlossen sein muß, wie man das schon von Zenec kennt. Vor der Ausgabe muß man aber noch entscheiden, ob ein Probeausdruck erfolgen soll. Wenn ja, gibt's den im Gegensatz zu Zenec aber nur in der normalen Schriftgröße. Menüpunkt 2 schreibt den Zeichensatz mit allen Druckersteuerzeichen auf Diskette. Dabei wird ein sequentielles ASCII-File erzeugt, welches ein "p" als Namensendung hat. Mit diesen Files ist es möglich, von Mastertext aus mehrere Zeichensätze in einem Text zu verwenden ohne den Ausdruck zu unterbrechen, denn solche ASCII-Files können mit der <ALT><L>-Funktion von Mastertext während des Ausdrucks direkt von der Diskette zum Drucker übertragen werden. Um zu demonstrieren wie die Zeichensatzänderung funktioniert, befindet sich ein Demo-Text ("MASTERF.DEMO K") auf der Diskette, der mit Mastertext und einem NEC P2200 ausgedruckt werden kann. Nach der Übertragung zum Drucker ist der neue Zeichensatz sofort aktiv (das gilt auch wenn der Zeichensatz direkt von Masterfont zum Drucker übertragen wurde).

Bei der Ausgabe auf Drucker und Diskette wird der gerade ausgegebene Buchstabe gleichzeitig auf dem Bildschirm angezeigt. Wenn man Menüpunkt 3 wählt, wird der Zeichensatz auf dem Bildschirm gezeigt und zwar als Hires-Grafik auf dem 80-Zeichen-Bildschirm. Bei der verwendeten Auflösung von 640 x 200 Punkten wird der Zeichensatz in zwei Teilen gezeigt, wobei zwischendurch auf einen Tastendruck gewartet wird. Nach der Ausgabe erscheint ein weiteres Menü auf dem Bildschirm, in dem man wieder mit den Tasten "1" bis "3" einen Menüpunkt wählen kann. Mit Menüpunkt 1 kann ein neuer Zeichensatz geladen werden. Nach der Wahl von Menüpunkt 2 erscheint wieder das oben beschriebene Ausgabe-Menü, und Menüpunkt 3 beendet das Programm. (aw)

Kurzinfo: Masterfont

Programmart: Assembler
Bildschirmmodus: 80 Zeichen (VDC)
Laden: "MASTERFONT" DLOAD
Starten: RUN
Besonderheiten: NEC-P 2200 benötigt
Benötigte Blocks: 12
Programmautor: Peter Hülstede

Kurzinfo: Zenec V2.0

Programmart: Assembler
Bildschirmmodus: 80 Zeichen (VDC)
Laden: "ZENEC V2.0" DLOAD
Starten: RUN
Besonderheiten: Zeichensätze sind teilweise sehr speicherintensiv
Benötigte Blocks: 33 (nur Hauptprogramm)
Programmautor: Hannes Müller

So finden Sie die Programme auf der Diskette

DISKETTE SEITE 1

```
Ø "128er-sonderh.89" s1 ca
Ø    "----------------" usr      66   "burstmon"          prg S. 40   Ø     "---------------"  usr
Ø    "---floppy 1571--" usr      3    "burstmon ass"      prg         119   "umrechnen"        prg S. 11
Ø    "----------------" usr      Ø    "----------------"  usr         Ø     "---------------"  usr
4    "dos copy.tabinst" prg S. 36 Ø   "--anwendungen---"  usr         Ø     "--erweiterung---" usr
51   "dos copy"         prg      Ø    "----------------"  usr         Ø     "---------------"  usr
6    "dos copy ass."    prg      67   "codiman"           prg S. 4    45    "Masterbasic.Inst" prg S. 26
3    "dos copy tab."    prg      2Ø   "codiman.list"      prg         112   "Masterbasic 128"  prg
3    "dc.ASCII"         prg      Ø    "----------------"  usr         4     "pcopy 8.obj"      prg
3    "dc.Mastertext"    prg      9Ø   "datenmanager 128"  prg S. 8    4     "pcopy24.obj"      prg
3    "dc.Startexter"    prg      1    "u-prg 1"           prg         Ø     "---------------"  usr
3    "dc.Vizawrite"     prg      7    "u-prg 2"           prg         Ø     "----diskette----" usr
3    "dc.Textomat"      prg      17   "zeichens"          prg         Ø     "---beidseitig---" usr
3    "dc.Textomat+"     prg      1    ".adressen demo"    prg         Ø     "----bespielt----" usr
Ø    "----------------" usr      3    "adressen demo"     seq         Ø     "---------------"  usr
```

DISKETTE SEITE 2

```
Ø "128er-sonderh.89" s2 ca
Ø    "-----------------"  usr           11    "auto-boot 128"     prg S. 49
Ø    "-tips & tricks---"  usr           Ø     "-----------------" usr
Ø    "-----------------"  usr           1     "mauscursor"        prg S. 48
33   "zenec v2.Ø"         prg S. 14     Ø     "-----------------" usr
44   "fraktur/h2Ø    .a"  prg           6     "moving colors 8Ø"  prg S. 48
44   "english/h2Ø    .a"  prg           Ø     "-----------------" usr
12   "masterfont"         prg           84    "lc-1Ø color"       prg S. 45
9    "masterf.demo    k"  prg           Ø     "-----------------" usr
3    "masterf.demo    d"  prg           12    "disketti i/new"    prg S. 49
2    "masterf.demo    f"  prg           8     "disketti m/new"    prg
25   "english/h2Ø .a/p"   seq           25    "disketti.d"        prg
3Ø   "small       .c/p"   seq           1     "disketti.mc"       prg
3Ø   "small.quer  .c/p"   seq           Ø     "-----------------" usr
44   "small          .c"  prg           153   "disk-hilfen"       prg S. 50
44   "small.quer     .c"  prg           Ø     "-----------------" usr
Ø    "-----------------"  usr           12    "defaults"          seq S. 44
9    "patch music"        prg S. 47     Ø     "-----------------" usr
Ø    "-----------------"  usr           Ø     "-------ende------" usr
                                        Ø     "-----------------" usr
```

WICHTIGE HINWEISE

zur beiliegenden Diskette:

Aus den Erfahrungen der bisherigen Sonderhefte mit Diskette, wollen wir ein paar Tips an Sie weitergeben:

1

Bevor Sie mit den Programmen auf der Diskette arbeiten, sollten Sie unbedingt eine Sicherheitskopie der Diskette anlegen. Verwenden Sie dazu ein beliebiges Kopierprogramm, das eine komplette Diskettenseite dupliziert.

2

Auf der Originaldiskette ist wegen der umfangreichen Programme nur wenig Speicherplatz frei. Dies führt bei den Anwendungen, die Daten auf die Diskette speichern, zu Speicherplatzproblemen. Kopieren Sie daher das Programm, mit dem Sie arbeiten wollen, mit einem File-Copy-Programm auf eine leere, formatierte Diskette und nutzen Sie diese als Arbeitsdiskette.

3

Die Rückseite der Originaldiskette ist schreibgeschützt. Wenn Sie auf dieser Seite speichern wollen, müssen Sie vorher mit einem Diskettenlocher eine Kerbe an der linken oberen Seite der Diskette anbringen, um den Schreibschutz zu entfernen. Probleme lassen sich von vornherein vermeiden, wenn Sie die Hinweise unter Punkt 2 beachten.

ALLE PROGRAMME
aus diesem Heft



HIER



Markt&Technik
Verlag Aktiengesellschaft

Diskette zum
Sonderheft
Nr. ____________

Diese Diskettentasche besteht aus chlorfrei gebleichtem Papier

Chefredakteur: Georg Klinge (gk) – verantwortlich für den redaktionellen Teil
Stellv. Chefredakteur: Arnd Wängler (aw)
Textchef: Jens Maasberg
Redaktion: Harald Beiler (bl), Heinz Behling (hb), Peter Klein (pk), Jörn-Erik Burkert (lb), Hans-Jürgen Humbert (jh)
Producer: Andrea Pfliegensdörfer
Redaktionsassistenz: Helga Weber

So erreichen Sie die Redaktion:
Tel. 089/4613-202, Telefax: 089/4613-5001, Btx: 64064

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträgern. Mit Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß die Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Layout: Isabell Schröfl, Erich Schulze
Fotografie: Roland Müller
Titelgestaltung und -grafik: Wolfgang Berns

Anzeigenleitung: Peter Kusterer
Anzeigenverwaltung und Disposition: Christopher Mark (421)

Anzeigen-Auslandsvertretung:
Großbritannien und Irland: Smyth International, Telefon 0044/81340-5058, Telefax 0044/81341-9602
Niederlande und Belgien: Insight Media, Telefon 0031/21531 2042, Telefax 0031/21531 0572
Italien: Medias International, Telefon 0039/31751494, Telefax 0039/31751482
USA und Kanada: M & T International Marketing, Telefon 001/415358-9500, Telefax 001/415358-9739
Japan: Media Sales Japan, Telefon 0081/33504-1925, Telefax 0081/33595-1709
Taiwan: Acer TWP Corporation, Telefon 00886-2-7136959 Telefax 00886-2-7151950
Korea: Young Media Inc., Telefon 0082-2-7564819, Telefax 0082-2-7575789
Israel: Baruch Schaefer, Telefon 00972-3-5562256, Telefax 00972-3-5566944
International Business Manager: Stefan Grajer 089/4613-638

So erreichen Sie die Anzeigenabteilung:
Tel. 089/4613-962, Telefax: 089/4613-791

Leiter Vertriebsmarketing: Benno Gaab

Vertrieb Handel: MZV, Moderner Zeitschriften Vertrieb GmbH & Co. KG, Breslauer Straße 5, Postfach 11 23, 8057 Eching, Tel. 089/319006-0

Verkaufspreis: Das Einzelheft kostet DM 24,-

Leitung Technik: Wolfgang Meyer (887)

Druck: SOV. Graphische Betriebe, Laubanger 23, 8600 Bamberg

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Artikel können für Werbezwecke in Form von Sonderdrucken hergestellt werden. Anfragen an Klaus Buck, Tel. 089/4613-180, Telefax 089/4613-232

Vorstand: Carl-Franz von Quadt (Vorsitzender), Dr. Rainer Doll, Dieter Streit

Verlagsleiter: Wolfram Höfler
Produktionschef: Michael Koeppe

Direktor Zeitschriften: Michael M. Pauly

Anschrift des Verlags: Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Die Zeitschrift wird mit chlorfreiem Papier hergestellt.

ISSN 0931-8933

Copyright-Erklärung

Name:...

Anschrift:...

Datum:...

Computertyp:...

Benötigte Erweiterung/Peripherie:...

Datenträger: Kassette/Diskette..

Programmart:...

Ich habe das 18. Lebensjahr bereits vollendet

...., den ....

....
(Unterschrift)

Wir geben diese Erklärung für unser minderjähriges Kind als dessen gesetzliche Vertreter ab.

...., den ....

....

....

Bankverbindung:

Bank/Postgiroamt:...

Bankleitzahl:...

Konto-Nummer:...

Inhaber des Kontos:...

Das Programm/die Bauanleitung:..

das/die ich der Redaktion der Zeitschrift 64'er übersandt habe, habe ich selbst erarbeitet und nicht, auch nicht teilweise, anderen Veröffentlichungen entnommen. Das Programm/die Bauanleitung ist daher frei von Rechten anderer und liegt zur Zeit keinem anderen Verlag zur Veröffentlichung vor. Ich bin damit einverstanden, daß die Markt & Technik Verlag AG das Programm/die Bauanleitung in ihren Zeitschriften oder ihren herausgegebenen Büchern abdruckt und das Programm/die Bauanleitung vervielfältigt, wie beispielsweise durch Herstellung von Disketten, auf denen das Programm gespeichert ist, oder daß sie Geräte und Bauelemente nach der Bauanleitung herstellen läßt und vertreibt bzw. durch Dritte vertreiben läßt.

Ich erhalte, wenn die Markt & Technik Verlag AG das Programm/die Bauanleitung druckt oder sonst verwertet, ein Pauschalhonorar.

Die Public-Domain-Szene von CP/M

CP/M-Programme (fast) zum Nulltarif

Prima. CP/M-3.0-Plus läuft: DIR, PIP, SHOW ... und das soll alles gewesen sein? Mitnichten! Ohne finanziellen Erdrutsch gibt's jede Menge Software – man muß nur wissen, wo.

Echte Software-Dorados sind z.B. Computer-Clubs. Schon zu Gründerzeiten der Home-Computer (oder was man damals dafür hielt) gab's Interessengemeinschaften: Special Interest Groups (SIG). Die bis dato größte und bekannteste ist die SIG for Microcomputers SIG/M, deren Bibliothek ca. 300 Disketten umfaßt (Programme und Dokumentationen, Quellcodes, Spiele und Anregungen für Computer mit 8080-, 8085-, Z80- und 65xx-Mikroprozessoren).

Auch heute noch präsent: die NAOG/ZSIG, eine 1988 entstandene, Z80-, Z180- und Z280-orientierte Programmierer-Gruppe.

Hier führt man konsequent den SIG/M-Gedanken mit Schwerpunkt Z80-kompatible Prozessoren weiter: Die Konfigurationsvoraussetzung der meisten PD-Programme ist ZCPR, eine Betriebssystemerweiterung für CP/M, die jede Menge mehr Komfort bietet und viel Arbeit erspart. Darüber später mehr.

Nicht mehr ganz so aktiv ist die FOG (First Osborne Group), deren umfangreiche CP/M-Sammlung (über 200 Disketten) C-128- und KayPro-Anwendern eine Riesenauswahl bietet. Leider gibt der Club nur vollständige Disketten an Mitglieder weiter. Die individuelle Verteilung einzelner Dateien ist von Privatperson zu Privatperson also leicht möglich, für Distributeure oder Vertriebsfirmen aus Zeitgründen aber nicht machbar.

Die Piconet Group, deren knapp 50 Disketten die mit Abstand beste MBASIC-Bibliothek enthalten, hat bereits das Handtuch geworfen. Auch für 8080- und 8085-User war's eine sprudelnde Quelle nützlicher, kleiner Utilities. Das gleiche Schicksal ereilte die CPMUG (CP/M User Group), früher eine Truppe mit zahlreichen Mitgliedern. Diverse Programme der ca. 90 Disketten dieses Clubs wurden in den Software-Bestand der SIG/M integriert.

Dagegen ist die CUG (C User'sGroup) noch äußerst vital, allerdings mehr in Richtung High-Level-Rechner (PC/ATs). Dennoch: Hier findet man immer wieder brauchbare Quelltexte zu BDS-C, dem meist verbreiteten CP/M-C-Compiler oder für andere C-Dialekte. Das CUG-Team entwickelte u. a. auch die Quasi-Float-Unterstützung von Small-C, der einzigen Public-Domain-C-Version. Daraus entwickelte sich z.B. die Float-Unterstützung der BDS-C Version 1.6 c (auf dem derzeit aktuellsten Stand). Sie gibt's nur als Komplettpaket mit zwei Compilern (einer für CP/M, der andere zusätzlich für ZCPR mit allen Libraries) und dem berühmten RED-Editor, den man nach eigenen Wünschen konfigurieren kann.

Wie kommt man an CP/M-Software?

Noch vor wenigen Jahren gab's Public-Domain-Programme (Abb. 1 bis 3) an jeder Software-Ecke, heute muß man wie ein Spürhund danach suchen.

Die billigste Variante ist, sich die gewünschten Programme von einem Freund zu besorgen. Allerdings hat man keine Garantie, daß dieser (oder der Vorbesitzer der Software!) nicht irgendwann schon an den Daten herumgepfriemelt hat und die Programme dann möglicherweise nicht mehr korrekt laufen. Die SIG/M benutzt daher ein CRC-File bzw. die Datei CATALOG.# # # (# = jeweils eine Ziffer). CRC prüft bereits beim Aufruf, ob die Daten manipuliert wurden. Manche Händler ändern Programme sogar absichtlich, um z.B. antiquierte Packverfahren durch moderne zu ersetzen. Allerdings ist hier gewährleistet, daß die geänderte Software nach wie vor läuft. CRC bedeutet: Cyclic Redundancy Check – alle Teile einer Datei werden überprüft.

Für fortgeschrittenere User ist das »Downloaden« aus Mailboxen ebenfalls eine beliebte Methode, an CP/M-Programme zu kommen. Da es aber PD-Disketten mit ca. 80 bis 270 KByte Software bereits für drei Mark gibt, lohnt es kaum noch, Programme aus Mailboxen abzurufen.

Ein Großteil hierzulande angebotener Public-Domain-Software besitzt deutsche Anleitungen (als READ.ME-Datei oder Textausdruck). So überarbeitete Software kostet bis zu 30 Mark pro Diskette. Man bekommt sie z.B. in der Kotulla-Bibliothek (Firma Weeske, Backnang), dann aber überwiegend im Diskformat des Schneider CPC, also entweder auf 3-Zoll-Disketten oder im 80-Spur-Vortex-Format (5¼- bzw. 3½-Zoll).

Mehr Speicherplatz durch Komprimieren

In der Public-Domain- und Shareware-Szene stößt man immer wieder auf Dateien, die aus Platzgründen komprimiert oder gepackt wurden.

Das erste Kompressionsverfahren war »Squeezing«, das Dateien mit der Endung ».?Q?« erzeugte. Diese Files ließen sich mit dem CP/M-Utility USQ.COM entpacken; noch komfortabler ging's mit NSWEEP.COM, das ab Version 2.0 auch die Squeeze-Funktion enthält. Für CP/M Plus sollte man jedoch nur die Versionen 2.7 oder 2.7a verwenden, da sie den verfügbaren Diskettenspeicher korrekt ausgeben – auch beim Betrieb von Festplatten.

Ebenfalls aus dieser Zeit stammt die Dateiform .LBR. Solche Files enthielten ursprünglich Libraries (Bibliotheken, sinngemäß unkomprimierte Archivdateien, Texte und Programme), später in gesqueeztem Zustand. Diese LBRs wurden per LU (Library Utility) erzeugt. Es wurde durchs NULU (New Library Utility) abgelöst. Version 1.52 ist die aktuellste Fassung dieses Pack-Managers.

Neuere Pack-Algorithmen (Huffman-Codierung und Crunching) fingen an, sich durchzusetzen: Der ursprünglich beim Squeezen erreichte Speicherplatzgewinn von 10 bis 15 Prozent ließ sich jetzt auf maximal 80 Prozent steigern (Lempel-Zev-Welch-LZW-Crunching von Texten)! Kombinierte, intelligente Packer erschienen in der UNIX-Szene und setzen ihren Siegeszug in der MS-DOS-Welt fort.

Dabei fiel auch was für CP/M ab. Die einzige Version mit

Cruncher und Entpacker ist die ARC-Version 2.0, von Tilmann Reh für CP/M Plus optimiert: Datum und Uhrzeit einer Datei werden mitgespeichert.

Das Entpacken selbst läßt sich mit dem kleinen, aber schnellen Assembler-Programm UNARC.COM erledigen, das sogar mit PKARC erzeugte, gesquashte Files entpackt. Squashing ist ein Merkmal moderner ARC-Programme.

Bei MS-DOS und UNIX existieren noch die Packer ZIP und ZOO, Compress, ARJ und LHA(RC). Allerdings: Nicht jedes Format läßt sich mit CP/M entpacken, geschweige denn crunchen:

- ZIP-Files, gepackt mit PKZIP bis Version 1.1, kann man mit UNZIP.COM Version 1.5 und höher auspacken. Bei höheren PKZIP-Versionen geht bislang noch nichts.
- ZOO-Files lassen sich per BOOZ.COM entpacken, für ARJ-Files greift man auf CPMUNARJ.COM zurück.
- Public-Domain-Libraries neuester Prägung liegen meist im ARC-Format vor oder in (neuen) LBRs, die auch Dateien in gepackter Form enthalten. Diese Files lassen sich mit NULU dann lediglich entpacken, bleiben aber gecruncht. Dann brauchen Sie ein zusätzliches Utility: UNCRUNCH.COM (am besten eine Version höher als 2.8), die auch LZH-(Lempel-Zev-Huffman-)codierte Files entpackt.

Sinnvoll und sehr praktisch ist das Utility QL41.COM, mit dem das Disketten-Directory numeriert am Bildschirm erscheint. Alle Dateien lassen sich risikolos untersuchen – per Tipp auf die entsprechende Zifferntaste. Ist es keine Textdatei, schaltet das Programm automatisch in den Hexadezimal-Modus um (wie bei SID oder DDT). Ist die eingegebene Ziffer die Kennzahl einer LBR-Datei, kann man diese gegebenenfalls öffnen. Hierarchisch geordnet erscheint jetzt das Verzeichnis der Archivdatei, das sich ebenfalls sichten läßt. Möchte man eine oder mehrere Files entpacken, drückt man <E>. Beachten Sie, daß QL41.COM stets auf das Laufwerk zugreift, von dem es aufgerufen wurde. Beim C-128 mit der RAM-Disk ist es dann sinnvoll, nach M: zu wechseln und z.B. mit

```
M> QL41 B:
```

Laufwerk B: aufzurufen. Handelt es sich nur um Texte, ist es manchmal zweckmäßig, die Dateien gar nicht zu entpacken, sondern in eine Library zu lesen. Das geht aber nur, wenn der Textumfang den verfügbaren Arbeitsspeicher (ca. 32 KByte) nicht übersteigt. Erscheint die Meldung »Entire text doesn't fit into memory« dann entpackt man besser und liest den Text anschließend per NSWEEP (mit Druckmöglichkeit).

Was bietet Public-Domain-Software?

CP/M ist ein Betriebssystem, das auf physikalisch völlig unterschiedlichen Computern läuft. Die meisten PD-Programme wurden für CP/M 2.2 entwickelt, vor allem Utilities, die Disketten direkt manipulieren oder freien Speicherplatz berechnen. Achtung: Solche Programme lassen sich mit CP/M Plus nur bedingt einsetzen – man muß mit erheblichen Datenverlusten rechnen, wenn man solche Programme unter CP/M 3.0 auf eine Disk losläßt!

Um z.B. solche Mißverständnisse zu vermeiden, sind fast alle Programme mit einem Hilfstext versehen, als File-Bezeichnung dient der Programmname mit der Endung (Extension) .DOC; deutsche Anleitungen heißen dann entsprechend .DOK. Solche Dateien enthalten normalerweise eine Kurzanleitung und wichtige Hinweise oder Warnungen vor falschem Gebrauch. Bei Festplatten unter CP/M Plus sollte man stets vorsichtig sein, wenn einzelne Partitionen größer als 8 MByte sind. Nahezu alle CP/M-Utilities berücksichtigen die für CP/M 2.2 gültige Höchstgrenze. Setzt man sie bei größeren Partitionen ein, sind Crashes vorprogrammiert.

Das kleinere Übel sind Programme, die einfach nicht laufen wollen (sie wurden entweder nur für einen bestimmten

```
+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
+                                                           +
+                    J u g g ' l e r - 1 2 8                +
+                                                           +
+                   Demonstration Version                   +
+                     by M. Garamszeghy                     +
+                                                           +
+               (c) 1988 Herne Data Systems Ltd.            +
+                                                           +
+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +

This program provides a demonstration of some of the features
of the powerful Jugg'ler-128 utility for the C-128 in CP/M
mode with a 1570, 1571 or 1581 disk drive.  Jugg'ler provides
read, write and format support for more than 130 types of MFM
CP/M disks in an easy to use menu form.

NOTE:  This program is copyrighted material.  However, user groups
       and individuals may freely copy and distribute this demonstration
       on the condition that it MAY NOT BE MODIFIED IN ANY MANNER and/or
       SOLD SEPARATELY FOR COMMERCIAL GAIN.

Press a key to continue...█
```

[1] Akzeptiert mehr als 170 verschiedene CP/M-Diskettenformate: der Jugg'ler-128 (Demo von der Katalog-Disk)

```
A: FORMAT   COM : SCREEN40 COM : PIP      COM : READ     ME
A>dir
A: ADAM     COM : FIXTRK0  COM : JUGG     COM : JUGG     DAT : JUGG     DEF
A: JUGGNOTE DOK : JUGGUPD  DOK : README   DOC : S8       COM : SS1581   COM
A: UPDATE   DOC : XDIR     COM :
A>xdir.com
** XDIR v3.7  works on CP/M 3.0 **

.   0k
ADAM    .COM  2k
FIXTRK0 .COM  2k
JUGG    .COM 16k
JUGG    .DAT 22k
JUGG    .DEF  2k
JUGGNOTE.DOK  4k
JUGGUPD .DOK  6k
README  .DOC  4k
S8      .COM  2k
SS1581  .COM  2k
UPDATE  .DOC  6k
XDIR    .COM  2k

Disk A: Size=  334K with 13 Files  Used=  394K   Space=    0K
A>█
```

[2] Professionelle Ausgabe der Directory einer CP/M-Disk: XDIR.COM (im Vergleich zur Standardfunktion)

```
0100  C3 4B 02 00 00 FF 42 44  20 76 30 34 20 2D 20 31  .K....BD v04 - 1
0110  31 20 4D 61 79 20 38 37  20 2D 20 70 75 74 20 75  1 May 87 - put u
0120  73 65 72 20 6E 75 6D 62  65 72 20 69 6E 20 30 31  ser number in 01
0130  30 33 68 2C 20 46 46 20  69 6E 20 30 31 30 35 68  03h, FF in 0105h
0140  20 74 6F 20 73 68 6F 77  20 70 72 6F 67 72 65 73   to show progres
0150  73 2C 20 30 20 74 6F 20  73 75 70 70 72 65 73 73  s, 0 to suppress
0160  2E 20 20 50 72 6F 67 72  61 6D 20 66 69 6E 64 73  .  Program finds
0170  20 61 6E 64 20 6C 6F 63  6B 73 20 6F 75 74 20 61   and locks out a
0180  6E 79 20 62 61 64 20 62  6C 6F 63 6B 73 2C 20 74  ny bad blocks, t
0190  68 65 6E 20 73 68 6F 77  73 20 61 6E 79 20 66 69  hen shows any fi
01A0  6C 65 73 20 74 68 61 74  20 61 72 65 20 61 66 66  les that are aff
01B0  65 63 74 65 64 2E 20 20  49 72 76 20 48 6F 66 66  ected.  Irv Hoff
01C0  20 24 00 00 3A 05 01 B7  C8 3A C2 01 B7 C2 E8 01   $..:....:......
01D0  3A 32 07 FE 05 D8 CA E2  01 FE 0A D8 CA E2 01 FE  :2..............
01E0  0F D8 32 C2 01 C3 F6 01  3A C3 01 3C 32 C3 01 FE  ..2.....:..<2...
01F0  05 D8 AF 32 C3 01 3E 0D  CD 07 02 2A 32 07 CD 0E  ...2..>....*2...

BD04    .COM: Page 1 of 22  Cmnd or '?' for Menu:
```

[3] Spürt defekte Spuren und Sektoren von CP/M-Disketten auf: das Public-Domain-Programm BDO4.COM

Rechnertyp programmiert oder sind einfach schlecht). Man muß immer damit rechnen, daß sich auf vollbespielten PD-Disketten solche Blindgänger einschleichen.

Zu den meisten Programmen gibt's aber zusätzlich den Quelltext, der sich von geübten Freaks leicht ändern oder an den eigenen Rechner anpassen läßt. Interessierten Einsteigern bieten die Source-Codes vor allem einen Überblick, wie solche Programme funktionieren. Gerade zu DBase II, Pascal, C und BASIC findet man wahre Public-Domain-Juwelen!

Die meisten Anwender haben Schwierigkeiten mit der Terminalanpassung. Hauptsächlich die Steuerzeichen für »Bildschirm löschen« und die korrekte Plazierung des Cursors sind das A und O solcher Änderungen von Quell-Codes. Bei kompilierten Programmen geht's sowieso nur per Debugger.

File Patcher wie EDFILE oder ZPATCH (unter ZCPR) erlauben es, eine Datei zu laden und die Bytes im Speicher zu ändern. Das gilt auch für B29, Z8E und Disassembler wie DAZZLESTAR oder DASM bzw. DISZ80. Allerdings sollte man wissen, wie der eigene Computer solche Steuerzeichen erzeugt – oft hilft ein kurzer Blick ins Handbuch!

Erscheint bei derartigen Programmen gleich zu Beginn eine Copyright-Meldung von Borland, dann ist es mit an Sicherheit grenzender Wahrscheinlichkeit ein Turbo-Pascal-File. Und schon gibt's Probleme: Unerfahrene Programmierer haben z.B. beim Kompilieren die höchstmögliche Speicher-Endadresse angegeben. Folge: Das Programm kann nicht laufen, weil der verfügbare Speicher (TPA) zu gering ist. Hier hilft bei geBANKten CP/M-Versionen (wie CP/M 3.0) auch keine Speichererweiterung – in so einem Fall kann man Zusatzspeicher nur als RAM-Disk nutzen. Reicht der Arbeitsspeicher aber aus und gilt es nur, das Terminal anzupassen, läßt sich das kinderleicht mit dem PD-Programm PDTINS.COM bewerkstelligen.

Wer beim Scannen mit EDFILE und ZPATCH im gesamten Programm kreuz und quer verstreute Cursor-Befehle findet, kann fast sicher sein, daß es ein kompiliertes Basic-Programm ist: Änderungen sind praktisch undurchführbar (zu arbeitsaufwendig!). Wer aber alles so lassen will, wie's ist, findet vor allem für Kaypro-Computer ausgezeichnete Basic-Dateien, ja sogar einen Mini-Flugsimulator! Für C-128-CP/M-User äußerst interessant: Die Bildschirmsteuerzeichen der Kaypros sind dem Televideo-Modus ähnlich; die Programme laufen also mit CP/M 3.0 des C128 (selbstverständlich nur im 80-Zeichen-Modus!).

Überarbeitete CP/M-Version: Z3PLUS

Wer gerne neue Sachen ausprobiert, kann sich mit dem erweiterten CP/M-System Z3PLUS austoben (99 Mark). Für den C128 gibt's eine bereits optimierte Version, die sich automatisch ins CP/M-Plus einnistet. Das macht das Programm u.a.: In einem speziellen Terminal-Speicherbereich werden alle Bildschirmsteuerzeichen abgelegt. Typische ZCPR-Public-Domain-Programme holen sich dann diese Informationen selbständig aus dem Speicher. Deshalb sind zusätzliche Installationen unnötig – höchstens noch Kosmetik für bestimmte Optionen. Das ermöglicht auch dem Ungeübten den Umgang mit fremden Programmen und erleichtert ihm sogar die tägliche Arbeit mit CP/M! Viele Tools unterstützen C-128-Blockgrafik: ästhetische Bildschirmausgaben also auch mit CP/M!

Die vom Vertreiber kostenlos über Unterverteiler abgegebene PD-Software hat mittlerweile einen Umfang von mehr als 40 MByte (= 40 Millionen Byte) erreicht, erhältlich in gecrunchter Form auf etwa 64 Disketten mit jeweils ca. 700 KByte. Übrigens: Der Kreis begeisterter Z3PLUS-User wächst von Monat zu Monat. Langeweile ist ab sofort ein Fremdwort: Allein das Sichten der Utilities dauert schon Wochen.

Zum Lieferumfang von Z3PLUS gehören zwei Kaypro-IV-Disketten und ein deutsches Handbuch. Die PD-Disketten kann man nur im User-to-User-Tausch bekommen: Ehrenamtliche Unterverteiler werden regelmäßig mit neu erscheinenden Scheiben versorgt, die wiederum gegen geringe Gebühr (quasi die Selbstkosten!) an die Z3PLUS-Anwender verschickt werden. Voraussetzung: Der interessierte User muß seiner Bestellung die entsprechende Anzahl CP/M-formatierter Disketten, sowie einen ausreichend frankierten Rückumschlag beifügen.

Z3PLUS gibt's bei Helmut Jungkunz, Zacherlstr. 14, 8045 Ismaning, 99 Mark + Nachnahmekosten. Die genannten CP/M-PD-Programme: selbe Adresse. Gegen Einsendung einer Kaypro-IV-formatierten Leerdisk und entsprechendes Rückporto kann man eine CP/M-Katalog-Diskette anfordern, die einen umfassenden Überblick über weitere CP/M-Software gibt (überwiegend Public Domain).

Kleines CP/M-Glossar

Begriff	Erklärung
Utility	kurzes, separates Hilfsprogramm für einen bestimmten Zweck (z.B. Diskmanipulationen, Kopieren usw.). Beispiele: USQ.COM, XD.COM
Tool	leistungsfähiges Werkzeug zur Bearbeitung von Daten, oft auch in kommerzieller Umgebung. Beispiele: NSWEEP.COM, NULU.COM
README, READ.ME	Dateien als reiner ASCII-Text, durch die Programmierer und Lieferanten dem Empfänger eine wichtige Mitteilung zukommen lassen. Solche Files werden mit der CP/M-Anweisung TYPE oder per Utility NSWEEP gelesen.
README.2, README.TOO	(»Liesmich.auch«), Zusatzinfos zu READ.ME oder den entsprechenden Programmen.
Extrahieren (engl. extract)	Entpacken einer Archivdatei
Sichten (engl. viewing)	Disketteninhalte oder Dateien nur lesen und ansehen.
Interpreter	Code einer Programmiersprache wird während des Lesens ausgeführt. Langsame Ablaufgeschwindigkeit.
Semi-Compiler	Zwischen-Compilate, erleichtern und beschleunigen die Ausführung.
Compiler	erzeugt schnellen Maschinencode aus einem Programmiersprachen-Quellcodes
Linker	Von Compilern erzeugte Objekt-Codes werden zu lauffähigen COM-Dateien gelinkt: notwendige, zusätzliche Teile aus sog. LIBs werden eingebunden und das Modul zur eigenständigen, selbstablaufenden Applikation.
ZCPR	Der Standard-CP/M-Kommando-Interpreter des CP/M wurde seinerzeit in langsamem 8080-Code programmiert. ZCPR ist eine konvertierte Version, die Programm-Code des Z80-Mikroprozessors verwendet. Vorteil: Es bleibt jetzt noch genügend Platz für Extras: z.B. genormte Terminal-Umgebung ohne Installation, benannte Directories und Zugriff auf alle User-Bereiche.
Z3PLUS	Autoinstallationsversion für CP/M Plus in Verbindung mit ZCPR. Semi-kommerzielles Produkt mit großem PD-Software-Bestand.
Terminal	ursprünglich ein Gerät, bestehend aus Bildschirm und Tastatur mit eigener Bezeichnung. Bei den meisten CP/M-Rechnern dienen Terminals über die serielle Schnittstelle zur Ein- und Ausgabe von Text (daraus abgeleitet: die heutige Hauptbedeutung »Summe der Bildschirmsteuerzeichen eines Computers«). Bekannte Terminaltypen: VT240, VT200, VT100, VT52, ANSI, TVI912 bis 955, WYSE, ADM3A und ADM31. Bemerkenswert: Viele Computer (vor allem DOS-Rechner) emulieren bekannte Terminals.
Emulieren	Nachahmen eines bestimmten Vorgangs oder Verhaltens. Die interessantesten sind außer den für die Terminals die CP/M-80-Emulationen unter DOS und UNIX. Sensationell: MYZ80 z.B. erlaubt sogar den Betrieb von CP/M Plus (und natürlich Z3PLUS, einer CP/M-3-Variante des ZCPR34)!
Runtime-Modul	zwischengeschalteter Spezial-Interpreter, der Semi-Compilate liest und sie beschleunigt ausführt. Beispiel: JRT-Pascal oder BRUN von MBASIC.

Überarbeitete Software für CP/M

... aber bitte mit Format!

Längst ist sie versiegt, die CP/M-Software-Quelle von Commodore. Kein Problem – andere Vertriebsfirmen haben den Platz des Computer-Herstellers eingenommen. So haben wir z.B. ein Programm wiederentdeckt, das 170 verschiedene CP/M-Formate verarbeitet.

Der C128 verwaltet vier Commodore-eigene CP/M-Formate, ergänzt durch neun Standardformate. Erschwerend wirkt sich aus, daß es mindestens drei unterschiedliche CP/M-3.0-Plus-Versionen gibt: die vom Sommer 1985, Dezember 1985 und Mai 1987. Die erste Fassung unterstützte die serielle Schnittstelle, die zweite den Drucker (die serielle Schnittstelle hat man dabei aber unerklärlicherweise vergessen); die dritte akzeptierte die RAM-Floppies 1570/1571 und das seinerzeit brandneue 3½-Zoll-Laufwerk 1581.

Soll ein Programm auf dem C128 laufen, gibt's zwei Möglichkeiten:
- entweder greift es direkt auf die CPU 8502 zu – dann ist es schnell, funktioniert aber nicht mit dem Mikroprozessor Z80,
- oder es ist ein universelles CP/M-Programm, das entweder ADM31-, ADM3A- bzw. Televideo-Terminalsteuerung kennt. Solche CP/M-Software arbeitet auch mit dem Z80-Prozessor des C128 – vorausgesetzt, sie braucht nicht zuviel Speicher und das verwendete Diskettenformat ist lesbar. Da hilft nur der direkte Datenaustausch mit den entsprechenden Laufwerken:

– Floppy 1570: ...physikalisch einseitiges Laufwerk, 5¼ Zoll. Es eignet sich fürs KayPro-II-Format (immerhin 196 KByte Speicherkapazität) oder IBM-CP/M86 (acht Sektoren pro Spur, einseitig – nicht MS-DOS!). Das letztgenannte Standardformat besitzt viele CP/M-Rechner, z.B. die Amstrad/Schneider-CPCs. Nachteil: die geringe Kapazität von lediglich 153 KByte. Dann gibt's Osborne 1 Single Sided (SS), das aber mit seinen 5 x 1024 Sektoren pro Spur mit diversen Computern und Laufwerken nicht zusammenarbeitet. Das C-128-CP/M-Format ist nicht standardmäßig organisiert und läßt sich daher nur mit dem C128 lesen.

– Floppy 1571: ...zwei- bzw. doppelseitiges Laufwerk mit intelligentem Controller, aber bedeutend flotter als die 1570. Mit der 1571 läßt sich zusätzlich das Kaypro-IV-Format verwenden – mit 392 KByte eine stattliche Diskettenkapazität. Ebenso beliebt (wenn auch durch unterschiedliche Implementierung nicht ideal) ist das Format Epson-QX10. Achtung! Davon gibt's drei Varianten:
- 16 x 256-Byte-Sektoren je Spur (128 Einträge, 298 KByte),
- die europäische Variante, ebenfalls mit 256-Byte-Sektoren (aber nur 64 Einträge und 276 KByte),
- Epson Valdoc, 10 x 512-Byte-Sektoren (128 Einträge und 374 KByte),

– Floppy 1581: ...doppelseitiges 3½-Zoll-Laufwerk, aber unglücklicherweise mit invertierter Kopfseitennumerierung – also nicht nach Standard! Das Laufwerk kann also nur das Commodore-1581-CP/M-Format unterstützen.

Das Lesen der genannten Formate geht stets nach Schema F: Diskette einlegen, das CP/M 3.0 des C128 erkennt das Fremdformat und meldet es (zumindest, was es zu erkennen glaubt!). Drückt man nun <ENTER>, bestätigt man die Auswahl und die Diskette wird eingeloggt. Besitzt man eine 1571 mit einer Kaypro-IV-Disk, erscheint aber möglicherweise die Anzeige: Kaypro II. Das ist selbstverständlich nicht korrekt und muß per Cursor-Tasten berichtigt (Kaypro IV) und erneut mit <ENTER> bestätigt werden.

Auf diesen Grundregeln basiert ein ausgereiftes Software-Produkt, das es seit kurzem wieder in der Version 3.4 gibt: Jugg'ler-128.

Hinter diesem Namen verbirgt sich ein Programm, das 170 verschiedene Diskettenformate bearbeitet und diverse Konfigurationen unterstützt. Selbst invertierte Formate wie »Super-Brain« lassen sich verarbeiten.

Wie funktioniert Jugg'ler?

Sämtliche Diskettenparameter werden in Tabellen verwaltet (Disk-Parameterblocks). Doch nicht alle reservierten Tabellen des C-128-CP/M-Plus sind belegt: die drei frei gebliebenen benutzt Jugg'ler, das CP/M-System im Vorbeigehen zu patchen. Fremdformate lassen sich ebenso leicht einstellen wie integrierte. Dabei werden Einträge aus einer Art Datenbank extern gelesen und als kurze Definitionsdatei gespeichert. Damit beschleunigt sich das Nachladen. Es lassen sich nur maximal drei neue Formate (Disk-Typen) installieren. Ist ein Sonderformat dabei, ist unter Umständen gar bei zwei Schluß, da der Speicherbereich für zusätzliche Tabellen benötigt wird.

Zum Lieferumfang von Jugg'ler gehört eine deutsche Bedienungsanleitung. Die Programm-Disketten (im KayPro-II-Format) lassen sich von den Floppies 1570 und 1571 problemlos lesen.

Die neueste Version von Jugg'ler unterstützt auch ECMA-Formate, deren erster Sektor mit einfacher Schreibdichte (FM) formatiert ist, der Rest allerdings in doppelter (MFM). Die 1571 kann diese gemischte FM/MFM-Spur normalerweise nicht lesen (außer, man baut das Laufwerk hardwaremäßig um). Bequemer ist es, das Utility FIXTRK0.COM zu verwenden, wobei die 1571 als DEVICE 8 installiert sein muß (als virtuelles Laufwerk). »Virtuell« bedeutet, daß ein physikalisches Laufwerk außer mit angestammten Namen auch noch unter einer anderen Bezeichnung existiert und angesprochen wird, obwohl die logischen Eigenschaften (Verwaltung) durchaus anders sein können. FIXTRK0.COM formatiert nun diese Problemspur einheitlich mit 16 x 256 Sektoren. Für den Ursprungscomputer wird diese Spur damit eventuell unlesbar, mit Sicherheit läßt sich die Diskette nicht mehr BOOTen!

Doppelseitige Diskettenlaufwerke haben den Vorteil, daß abwechselnd ein Kopf von oben und einer von unten die Daten liest – in welcher Reihenfolge, das hängt vom Format ab. Standardmäßig sind diese Köpfe mit 0 und 1 numeriert und adressiert. Commodore wollte bei der 1581 aber offensichtlich ein paar Groschen für ein zusätzliches NAND-Gatter sparen – die Kopfnummern sind nämlich im Vergleich zur restlichen Computerwelt vertauscht! Besonders bei einseitigen Formaten entsteht dann der tolle Effekt, daß die Diskette aus dem Blickwinkel der 1581 mit der unbeschriebenen Seite zuerst ankommt und als unformatiert bzw. leer abgelehnt wird – Ende der Fahnenstange! Aus diesem Grund gibt's für den Jugg'ler noch ein Utility: SS1581.COM. Es veranlaßt die 1581, zuerst Seite 1 (statt 0) einzuloggen – dann läuft der Rest wie geschmiert.

Das Utility SB.COM liest invertierte Formate, bei denen die binäre Information als Komplement gespeichert ist: Nullen

werden zu Einsen und umgekehrt. Beispiel: Aus der Hexadezimalzahl $44 (binär 0100 0100, dez. 68) entsteht 1011 1011 (hex. $BB, dez. 187).

Für die Adam-Coleco-Computer existiert ebenfalls ein Tool: ADAM.COM (hierzulande wegen der geringen Verbreitung der Adam-Geräte bedeutungslos).

Die genannten Dienstprogramme sind »saubere« Utilities, die sich nach Gebrauch wieder aus dem Speicher entfernen. Außerdem verwenden sie RAM-Bereiche, deren Zutritt normalen Anwenderprogrammen verwehrt ist.

Ein nützlicher Menüpunkt ist die Analyse fremder Diskettenformate. Hierbei versucht Jugg'ler automatisch, den Diskettentyp zu erkennen. Selbstverständlich existiert auch eine Option zum Ausgeben der Formatliste, die sich per <CTRL P> zum Drucker schicken läßt.

Bei Jugg'ler kann man optionale Startparameter per Kommandozeile eingeben (Direktmodus) und dann ins CP/M zurückspringen. Ebenso wurde eine spezielle Patch-Funktion eingebaut: CCP.COM muß jetzt bei künftigen CP/M-Systemstarts nicht mehr nach dem Autostart-File PROFILE.SUB suchen, sondern nach JUGG.COM. Selbstverständlich läßt sich diese Funktion auch wieder abschalten.

Jugg'ler läßt diverse MFM-Maximalformate zu, die man dann als Standardeinstellung benutzen kann:

- MAXI 1571: (5 x 1024, 128 DIR, 396 KByte),
- MAXI 1581: (5 x 1024, 128 DIR, 796 KByte),
- MG 1581: (10 x 512, 128 DIR, 776 KByte).

Diese Formate werden wiederum von den Laufwerken solcher Computer erkannt, die ein spezielles BIOS besitzen (z.B. mit Selbstbau-ECB-Bus; Rechner von Reh Design, Siegen, deren CP/M Plus mit einer Z-280-CPU und 12 MHz arbeitet und die bis zu vier Laufwerke unterschiedlicher Dichte verwalten, sowie - über einen Zusatz-Controller - AT-Bus-Festplatten verfügen). Zum Vergleich: Controller eines PC/XT können nur maximal 360 KByte formatieren.

Fazit

Das Programm bietet die größte Auswahl an CP/M-Fremdformaten, die wir bislang kennen. Einer wahren Datenaustausch-Flut zwischen unterschiedlichen CP/M-Systemen steht damit nichts mehr im Weg. Lobenswert: Jugg'ler ist nicht kopiergeschützt. Der akzeptable Preis von 70 Mark gestattet selbst schmalen Geldbeuteln die Anschaffung. Treffender Werbespruch des Herstellers: Software worth hunting...

(bl)

Jugg'ler, CP/M-Diskettenformat: KayPro II/IV, deutsche Anleitung, Helmut Jungkunz, Zacherlstr. 14, 8045 Ismaning, 70 Mark

Jugg'ler (Gesamtliste der Diskettenformate)			
ABC - 80	Cromemco CDOS	Magic	Sharp MZ-80
AB Dick Magna III	Cromemco Int'l Term	MAXI 1571	Slicer
Access Matrix	Cromemco Int'l Term	MAXI 1581	Sony SMC-70 (1581)
Actrix	C/WP Cortex	MG 1581	Spectravideo SV-328
Actrix	Cykey	Mega-88 Turbo DOS	SuperBrain Jr.
Adler Textwriter	DEC Robin	MicroBee	SuperBrain 40 Track
Adv. Digt'l Super-6	DEC VT-180	Micron Quark	SuperBrain
Adv. Digt'l Super-6	DEC II/Rainbow (1581)	Micro Source M-6000	Systel 2
Adv. Digt'l T-DOS 312K	Digilog	Molecular S-9	Systel 3
Adv. Digt'l T-DOS 366K	Direct 1025	Morrow MD2	Tektronix 4170
Amigo	Epson QX-10 (256)	Morrow MD3,5,11,16,3	Telectronics CP/M 681
Amigo	Epson QX-10 (256-Eur1)	Morrow Turbo DOS	Teletek SystemMaster
Ampro	Epson QX-10 (512)	Mountain Control	TeleVideo 801, 806
Ampro (1581)	Formula 1	MS-DOS (8 sector)	TeleVideo Turbo DO
Archive	Fujitsu Micro 8	MS-DOS (8 sector)	TI Pro. CP/M-86
Arisia	Fujitsu Micro 16-s	MS-DOS (9 sector)	Tomcat 1800G
ATR-8000	Gemini (1581	MS-DOS (9 sector)	Toshiba T-100
ATR-8000	Glass Typewriter II	Multiflex	Toshiba T-200
Avatar	Glass Typewrtr (1581)	NCR Decision Mate	TRS-80 Model 1
Beehive	Gnat Sys. 10	NEC PC-8001	TRS-80 Model 1/Omicr1
Bitelex	Heath Magnolia	NEC PC-8801	TRS-80 Color Power I
Bondwell 12	Heath Magnolia	NEC PC-8431 (1581)	TRS-80 Mod 3/Holmes
Bondwell 14	Heath Magnolia (1581)	Nelma Persona	TRS-80 Mod 3/M.M.
Bondwell 2 (1581)	Hewlett Packard 86	Olivetti 240 (1581)	TRS-80 Mod 4/MTZ (25)
Burr-Brown	Hew.-Pack. 125 (1581)	Olivetti ETV-300	TRS-80 Mod 4/MTZ (25)
C-64 CP/M (GCR)	IBM-PC (CP/M-86)	Olivetti M20	TRS-80 Mod 4/MTZ (51)
C-128 CP/M (GCR)	IBM-PC (CP/M-86)	Olympia ETX-II	TRS-80 Mod 4/MTZ (51)
C-128 CP/M (GCR)	ICM Turbo DOS	Olympia EX-100	TRS-80 Mod 4 CP/M+
C-128 1581 CP/M	IMS-5000	Osborne 1	Wang-Writer
Cal PC	IMS-5000	Osborne 1/Executive	Wavemate Bullet
CCS (256 Byte/Sector)	Insight Enter.	Osborne G2	Xerox 16/8
CCS (512 Byte/Sector)	Jet-80	Osborne Nuevo/Vixen	Xerox 820
CCS (1024 Byte/Sector)	KayPro II, 2	Otrona Attache	Xerox 820-II
CDI-5000	KayPro IV, 10	Panasonic KXE-828	Xerox 820-II
Coleco ADAM CP/M	Labtam (1581)	Pegasus Data Logger	Xerox/Sunrise Laptop
Commodore DOS (GCR)	Lexoriter	Philips PC-2010	XOR-5
Compugraphic MCS 5	Librarian 1571	PMC MicroMate	Zenith Z-37
Compustar 30	Librarian 1581	Portapak (1581)	Zenith Z-90
Cromemco C-10	Lobo Max-80 (256 Byt1)	Royal Alphatronic	Zenith Z-89/90
Cromemco CDOS	Lobo Max-80 (256 Byt1)	Sage IV	Zenith Z-89 (1581)
Cromemco CDOS	Lobo Max-80 (512 Byt1)	Sanyo MBC-1000, 1150	Zenith Z-100
Cromemco CDOS	L.N.W.-2	Seequa Chameleon	Zenith Z-100
		Seequa Chameleon	Zorba

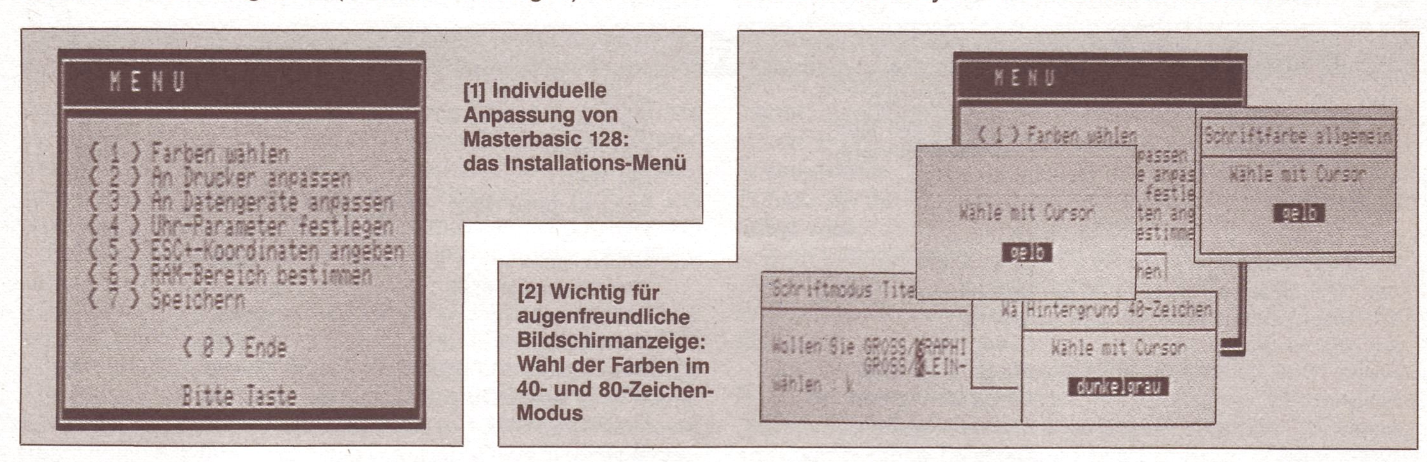
Masterbasic 128 – neue Dimensionen für Programmierer

Die höchste Stufe

Wer bisher geglaubt hat, das Basic 7.0 des C 128 sei der Weisheit letzter Schluß, sollte seine Meinung schnell revidieren: »Masterbasic 128« von Jürgen Hohmann bietet Anweisungen und Funktionen, die bislang Basic-Dialekten der 16- bzw. 32-Bit-Computer (PC/AT, Amiga, Archimedes usw.) vorbehalten waren.

Basic 7.0 wird mit 51 neuen Befehlen und 25 Funktionen aufgemotzt; 21 Originalanweisungen sowie die ESC-Taste erhalten zusätzliche Features.

Je nach gewünschter Speicherbank (0 oder 1) lädt und startet man das Programm (Wert für x eintragen):



[1] Individuelle Anpassung von Masterbasic 128: das Installations-Menü

[2] Wichtig für augenfreundliche Bildschirmanzeige: Wahl der Farben im 40- und 80-Zeichen-Modus

```
bank x: bload "Masterbasic 128":sys49152
```

Die Basic-Erweiterung kann man nicht aus laufenden Programmen aufrufen: Der Computer stürzt unweigerlich ab.

Die Erweiterung arbeitet mit einem 40- oder 80-Zeichen-Screen. Ist der VDC-Textschirm aktiv, schaltet das Programm automatisch den 2-MHz-Takt ein.

Konfiguration und Installation

Bevor man die Basic-Datei »Masterbasic.Inst« startet, muß Masterbasic 128 in Bank 1 vorhanden sein:

```
run "Masterbasic.Inst"
```

Nach Bestätigung einer Sicherheitsabfrage wird die Erweiterung zum Ändern nach Bank 0 kopiert. Dann meldet sich das Installations-Menü (Abb. 1). Die Optionen wählt man per entsprechender Zifferntaste:

<1> **Farben wählen:** Zunächst bestimmt man, ob die Einschaltmeldung des neuen Basic-Systems künftig in Groß- oder Kleinschrift erscheinen soll.

Anschließend lassen sich Rahmen- und Hintergrundfarbe für den 40-Zeichen-Screen bestimmen: per Cursor-Tasten auswählen und mit <RETURN> bestätigen (Abb. 2).

<2> **An Drucker anpassen:** Damit manipuliert man die beiden Epson-kompatiblen Treiber-Dateien »PCOPY 8.OBJ« und »PCOPY24.OBJ« (auf der Diskette zu diesem Sonderheft), die bei Bedarf geladen werden.

Für Grafikdruck lassen sich insgesamt vier verschiedene Sequenzen definieren. Eine Sequenz darf aus maximal fünf Byte bestehen. Sind's weniger, muß man mit $FF auffüllen. Falls man die Funktion nicht benötigt, müssen Sie fünfmal $FF eintragen. Es sind nur Hexadezimalzahlen erlaubt. Die fünf definierten Steuer-Bytes bestätigt man per <RETURN>. Vergleichen Sie die Druckersequenzen mit Ihrem Handbuch und ändern Sie sie, falls nötig (Abb. 3).

Startfunktion (erste Sequenz): ... schaltet entweder den Zeilenabstand ein oder das automatische Linefeed aus.

Grafik-Sequenz: ... bestimmt den Grafikdruck. Wichtig ist, daß bei 8-Nadel-Druckern – je nach Spaltenzahl – Grafik-Pixel zwei- bzw. dreimal gedruckt werden, bei 24-Nadlern drei- bzw. viermal (also die höchstmögliche Druckauflösung). Die Anzahl der Grafik-Bytes wird automatisch berechnet.

Next-Line (dritte Sequenz): ... schaltet zur nächsten Zeile um.

Abschlußfunktion: ... aktiviert wieder den Zeilenvorschub bzw. automatisches Linefeed für normale Textausgabe.

Achtung: Die Hardcopy-Routine zu den 24-Nadlern benutzt nur 16 Nadeln (also 16/180 Zoll eintragen!). Bei 8- bzw. 9-Nadel-Druckern sind's aber weiterhin 24/216 Zoll. Falls Ihr Drucker bei »Anzahl der Grafikbytes« High- und Low-Byte benötigt, muß die nächste Frage verneint werden.

Jürgen Hohmann

Informatikstudent, geboren am 5.9.1968, kam Ende 1983 erstmals mit einem Computer in Berührung: Es war der C 64 (inkl. Floppy 1541). Seit der Erstausgabe im März 1984 verfolgt er das 64'er-Magazin.

Zunächst interessierte er sich nur für Spiele. Doch nach kurzer Zeit reichte ihm das nicht mehr: Er wollte mehr aus dem Computer rausholen und vertiefte sich in Assembler-Programmierung (Hypra-Ass C 64): Der Entwicklung kurzer Utilities folgten die Analyse des C-64-Betriebssystems und umfangreicher Listings in Maschinensprache.

Nach dem Umstieg zum C 128 stellte er fest, daß die meisten bislang veröffentlichten Basic-Erweiterungen lediglich kleine Zusätze des normalen Basic 7.0 waren, die sich aber aufgrund ihrer Position im Speicher oft gegenseitig ausschlossen. In akribischer Kleinarbeit sammelte er Ideen zur Verbesserung des Basic 7.0, um sie dann möglichst platzsparend und variabel zusammenzusetzen. Das Ergebnis: Masterbasic 128.

Last but not least lassen sich für den Druckertreiber noch die OPEN-Parameter für den entsprechenden Linearkanal (z.B. Sekundäradresse 1) und die Standard-OPEN-Werte für die Befehle PSET, PPRINT, PLIST und PCLOSE bestimmen.

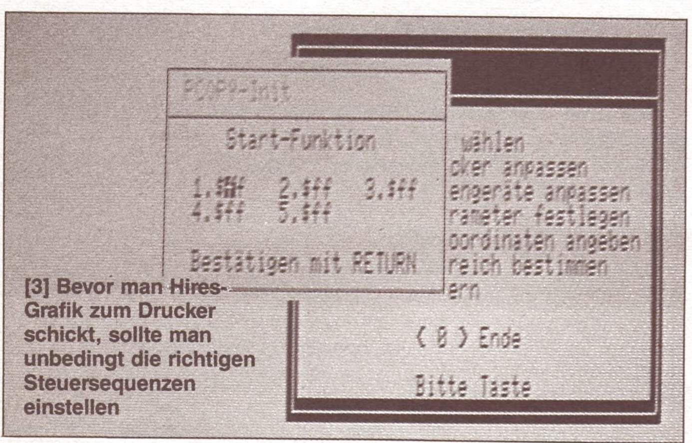
Wer die integrierte Software-Centronics-Schnittstelle verwenden will, kann die Geräteadresse und den Datenkanal für Groß- oder Kleinschrift bestimmen.

<3> **An Datengeräte anpassen:** Dieser Menüpunkt bezieht sich auf die Befehle DISK, SCNLOAD, SCNSAVE und START: Die jeweils voreingestellte Geräteadresse läßt sich damit ändern.

<4> **Uhr-Parameter festlegen:** Die eingebaute Interrupt-Uhr kann man digital auf beiden Bildschirmen zeigen. Zunächst legt man die Farben fest (40- und 80-Zeichen-Schirm), dann die Koordinaten, an deren Position die Uhr erschienen soll. Im VDC-Modus läßt sich dafür eine Extrazeile einrichten.

<5> **ESC + Parameter angeben:** Mit der Tastenkombination <ESC> < + > können Sie auf dem Bildschirm ein Fenster öffnen (Abb. 4). Für die Koordinaten gelten die Grenzen 0 < = xlo < xru < = 38 und 0 < = ylo < yru < = 23. Bei aktiviertem 80-Zeichen-Schirm werden beide x-Koordinatenwerte vor dem Window-Aufbau verdoppelt.

<6> **RAM-Bereich bestimmen:** Zunächst werden Sie gefragt, ob Masterbasic 128 im RAM oder in einem der ROMs



[3] Bevor man Hires-Grafik zum Drucker schickt, sollte man unbedingt die richtigen Steuersequenzen einstellen

installiert werden soll (muß vorher auf EPROM gebrannt werden!). Entscheidet man sich für die ROM-Version, gibt's zwei Möglichkeiten:

- im Bereich von $8000 bis $FFFF: Die Daten für den INFO-Befehl ($D000 bis $DFFF) werden nach $8000 kopiert.
- Installieren einer abgespeckten Programmversion von $C000 bis $FFFF. Dann muß man aber auf einige Befehle verzichten: KILL, CHANGE, UHR, CENTRONIC, ESCAPE, STRSAV, MEMORY, SCROLL, GCOPY, SCNSAVE, SCNLOAD, die Prozeduranweisungen, TRANS, FILL, MAXINP, LOCAL, GLOBAL, XDEF, VLOAD, VSAVE, SORT, LLIST, den erweiterten MONITOR und INFO.

Als internes ROM im vorhandenen Sockel liegt die Erweiterung in Bank 4, als externes in Bank 8. Mit »BANK x: SYS 49152« läßt sich dann die Erweiterung aktivieren.

Bei der RAM-Version entscheiden Sie, wo das neue Basic im Speicher liegt. Neben dem Befehlsnamen zeigt der Computer die unterste Speichergrenze. Per <CRSR aufwärts/abwärts> trifft man die Auswahl der gültigen Kommandos. »Minimumgrenze« ist das Limit, mit <RETURN> streichen Sie alle Befehle davor aus der Liste. Die RAM-Version läßt sich in Bank 0 oder 1 laden.

Ruft man später Anweisungen auf, die in der neu generierten Version unserer Basic-Erweiterung nicht enthalten sind, provoziert man die Fehlermeldung »UNIMPLEMENT COMMAND ERROR.

<7> **Speichern:** ... sichert Ihre individuell zusammengestellte Masterbasic-Version auf Disk. Dazu gibt man den Dateinamen sowie die Laufwerks- und Gerätenummern an.

<0> **Ende:** ...bricht das Programm nach einer Sicherheitsfrage ab.

Speicherbelegung

Die Erweiterung benützt den Speicherbereich von $9000 bis $FEFF (entweder Bank 0 oder 1). Die Speichergrenze der gewählten Bank stellt sich automatisch ein.

Die Common-Area wird ab $1000 definiert (statt $0400). Den RAM-Bereich von $0400 bis $0FFF in Bank 1 verwendet das Programm jetzt als Stapelspeicher für die Prozedur-Routinen. Im Bereich von $0C00 bis $0CFF wurden diverse Routinen und Parameter untergebracht, die als Schaltstelle zwischen dem Original-Betriebssystem bzw. Basic 7.0 und den neuen Befehlen fungieren. Ist die Centronics-Schnittstelle aktiv, braucht sie den Speicher von $0D00 bis $0D9C. Für temporäre Zwischenspeicher-Aktionen zu verschiedenen Befehlen weicht Masterbasic 128 auf den Kassettenpuffer aus (ab 0B00).

Die neuen Befehle von Masterbasic 128 (s. Textkasten) befähigen jeden Programmierer zu Super-Manipulationen. (bl)

```
1040 trans1,$9000,$feff,0,$9000
1050 bank0:doke$1212,$0000:scnclr
1060 ifrwindow(2)=40thenslow
1070 :
1080 pr
1090 re  1080 proc randomwindow(h,b,t$)
1100 :   1090 rem erzeuge ein fenster an zufaelliger stelle
1110 :   1100 : do:x=random(1,rwindow(2)-b-1):loopwhilex=peek($e6)
1120 :   1110 : do:y=random(3,23-h):loopwhiley=peek($e5)
1130 en  1120 : window x,y,x+b-1,y+h-1,131,t$
1140 :   1130 endproc randomwindow
1150 pr  1140 :
1160 re  1150 proc farbe(ad,y,m)
1170 :   1160 rem ermittle die bisherige farbe und frage die neue ab
1180 :   1170 : dimi,o,a$,f$(15),v%(15)
1190 :   1180 : lrestore farbtab
1200 :   1190 : fori=0to15:readf$(i),v%(i):next:i=peek(ad)
1210 :
1220 :   break
1230 :   ready.
1240 :
break
ready.
```

[4] Bildschirmfenster auf Tastendruck per ESC+-Funktion

Masterbasic 128 (Prozeduren-Priorität)

Priorität bedeutet hier: Es wird zuerst bei Priorität 1 gesucht, dann bei 2 usw. bis zum Programmanfang.

Unterprozeduren, die in Hauptprozeduren definiert wurden, sind von außen nicht erreichbar: Man kann also gleiche Prozedur-Namen verwenden.

	p1	p2	p3	p4	p5
m1	ja 2	ja 1	ja 2	nein	nein
m2	ja 1	ja 1	ja 1	nein	nein
m3	ja 2	ja 2	ja 1	ja 1	nein
m4	ja 3	ja 3	ja 2	ja 1	ja 1
m5	ja 1	ja 1	ja 1	nein	nein
m6	ja1	nein	nein	nein	nein
m7	ja 4	ja 4	ja 3	ja 2	ja 1

Kurzinfo: Neue Masterbasic 128

Programmart: Basic-Erweiterung
Bildschirmmodus: 40 und 80 Zeichen
Laden und starten: BANK 1: BLOAD "Masterbasic 128": SYS 49152
Besonderheiten: bietet 51 neue Befehle und 25 neue Funktionen. Die Anweisungen des Basic 7.0 gelten uneingeschränkt, Basic-Befehle lassen sich allerdings nicht mehr abkürzen.
Benötigte Blocks: 165
Programmautor: Jürgen Hohmann

Masterbasic 128 (Befehlsübersicht)

Grundlage dieser Basic-Erweiterung ist das im C128 integrierte Basic 7.0, dessen Befehle und Anweisungen nach wie vor uneingeschränkt gelten. 51 neue Befehle und 25 neue Funktionen sind dazugekommen, 21 existierende Anweisungen wurden ergänzt.

Masterbasic 128 läßt sich in Bank 0 oder Bank 1 installieren und per SYS DEC("C000") starten.

Geänderte Basic-7.0-Befehle

Die erweiterten Original-Anweisungen lassen sich im Direktmodus oder in Programmen nach IF, ELSE, BEGIN (ohne Doppelpunkt) aufrufen. Achtung: RREG liefert dann aber für die Register A, X und Y falsche Werte, da sich der Speicherbereich ständig verändert.

Falls nichts angegeben, lassen sich alle Parameter auch durch Variablen ersetzen. Werte in eckigen Klammern dürfen entfallen.

CATALOG [d drive][,u ga][,dateiname].

... zeigt das Directory der aktuellen Diskette im Laufwerk.

Der Unterschied zur normalen Anweisung: Der Bildschirm wird gelöscht, der Header (Diskettenname) scrollt nicht nach oben. Hinter dem zweiten Anführungszeichen erscheint ein Doppelpunkt - das vereinfacht das Laden von Dateien aus dem Inhaltsverzeichnis.

KEY [x, x$]/ KEY ON/ KEY OFF/ KEY OLD

Der Parameter x kann Werte von »1« bis »10« annehmen.

- KEY ohne Parameter listet die Belegung aller zehn Funktionstasten (inkl. <SHIFT RUN/STOP> = Key 9 und <HELP> = Key 10),
- KEY OFF ... schaltet die Funktionstastenabfrage ab: Die Tasten erhalten die vom C64 gewohnten ASCII-Werte (ab CHR$(133)),
- KEYON ... reaktiviert die Funktionstastenbelegung,
- KEYOLD ... setzt die Standardbelegung aller zehn Tasten.

MONITOR

... ruft den internen Maschinensprache-Monitor auf. Die neuen Funktionen:

- Das Status-Byte in der Registeranzeige zeigt sich übersichtlicher (Anzeige aller 8 Bit), kann aber nicht mehr verändert werden.
- Bei Zahlenumrechnungen per <$>, <+>, <&> und <%> kann man jetzt mit <+> oder <-> einen zweiten Wert anhängen und das Ergebnis auf den Screen bringen.

Beispiele:

```
$3 + %10
%1001 - +2
```

- *K [adr1 [adr2]]*: ... ähnelt der Anweisung »M« (Memory-Dump), bringt aber die Bytes als ASCII-Zeichen. Unidentifizierbare Werte erscheinen als Punkt <.>.
- *< adr xxxxx*: ... fügt die Zeichen »xxxxx« als ASCII-Code ab Adresse adr ein. Damit lassen sich Bytes überschreiben, die per K-Anweisung auf dem Bildschirm erscheinen. Zeichen, die Punkte <.> erzeugen, werden aber ignoriert - die ursprüngliche Bedeutung ändert sich nicht.
- *B [track sector [ga]]*: ... liest auf der Floppystation mit der Gerätenummer »ga« den Datenblock »sector« von Spur »track« in den Kassettenpuffer ($0B00 bis $0BFF) und zeigt ihn auf dem Screen. Blockinhalte kann man beliebig verändern. Fehlt der Parameter »ga«, nimmt das Programm automatisch Nr. 8.

Verzichtet man zusätzlich auf »track und sector«, holt die Funktion den nächsten logischen Diskettenblock in den Speicher. So hangelt man sich durch alle Sektoren eines Directory oder Daten-Files.

- *P [track sector [ga]]*: ... schreibt den Block $0B00 bis $0BFF auf die Diskette in der Floppystation »ga« zurück. Fehlen die Angaben zu »track sector«, gibt das Programm die aktuellen Werte aus: Man kann sie ändern oder per <RETURN> übernehmen.
- *U altanf altend neuanf anfchange endchange*: Wurde ein Assembler-Programm, das früher im Bereich von »altanf« bis »altend« lag, nach »neuanf« verschoben, ändern sich alle Kommandos zwischen »anfchange« und »endchange« - sofern sie sich auf den Bereich »altanf« bis »altend« beziehen.
- Die Anzahl der Zeilen und Spalten zu den Monitor-Befehlen D, M und K passen sich dem Bildschirmfensterformat an,
- bei der Bildschirmausgabe zum D-Befehl (Disassemble) erscheinen bis zu drei Byte (das Assembler-Mnemonic als ASCII-Zeichen),
- zu den Anweisungen M, D und K läßt sich nun auch eine Bildschirm-Hardcopy zum Drucker schicken (ESC-Codes werden eliminiert).

Beispiele:

- M adr1 adr2 :P ga sec
- D adr1 adr2 :P ga sec
- K adr1 adr2 :P ga sec

```
m 0b00 0bff: p 4 7
```

Voreinstellung: Geräteadresse (ga) = 4, Sekundäradresse (sec) = 0.

ON x GOTO/GOSUB – ON x CGOTO/CGOSUB – ON x LGOTO/LGOSUB label1:label2: ...

- ON x GOTO/GOSUB z1, z2 ...: Sprung zur Programmzeile x,
- ON x CGOTO/CGOSUB a+b, c*d ...: verzweigt zur Zeile, deren Nummer sich aus dem Rechenergebnis (z.B. a + b) bildet,
- ON x LGOTO/LGOSUB: ... ruft den Programm-Code auf, der ab dem mit »Label« bezeichneten Programmteil steht. Bei weniger als x Labels macht das Programm in der nächsten Zeile weiter.

WINDOW xlo, ylo, xru, yru[,lö][,titel$]]/ WINDOWOFF

... definiert ein Fenster auf dem aktuellen Bildschirm. Die x-Werte dürfen zwischen 0 bis 39 (bzw. 79 beim VDC) liegen, die y-Werte zwischen 0 bis 24. Die Parameterzahlen für Löschen (lö):

- lö = 0: Der Cursor rückt in die HOME-Position (oben links),
- lö = 1: ... löscht den Inhalt des Bildschirmfensters,
- lö = 2: ... definiert das Fenster, rahmt es ein und löscht es. Achtung: Jetzt dürfen die x-Werte nur zwischen 1 und 38 (78) liegen, y-Zahlen zwischen 1 und 23.
- lö = 3: ... bringt ein gerahmtes Fenster mit Kopfzeile (titel$, darf nur beim Parameter lö = 3 angegeben werden!). Die maximalen Koordinaten: x-Werte zwischen 1 und 38 (78), y-Werte zwischen 3 und 23.

Addiert man »128« zum Wert von lö, sichert man den alten Inhalt des Fensters inkl. Farb-RAM in Bank 0 (im freien RAM hinter Ihrem Basic-Programm). Also: je kleiner das Programm, desto mehr Windows lassen sich sichern, höchstens aber 255 - sonst erscheint die Fehlermeldung »TOO MANY WINDOWS«. Reicht der Restspeicher nicht mehr aus, erhält man »OUT OF MEMORY ERROR«.

WINDOW OFF stellt den ursprünglichen Fensterinhalt wieder her.

Zwischen Speichern und Lesen lassen sich beliebig viele Fenster ohne Sicherung definieren. Kommt der Bildschirmbereich nach WINDOWOFF nicht mehr korrekt auf den Monitor, muß man den Befehl SCNCLR aktivieren.

Beispiele:

```
window 10,4,30,10,2
```

... definiert ein umrahmtes Fenster,

```
window 0,0,39,24,128
```

... sichert entweder den gesamten 40-Zeichen-Bildschirm oder den halben 80-Zeichen-Screen. Der Inhalt bleibt originalgetreu erhalten und läßt sich mit WINDOW OFF wiederherstellen.

```
window 5,10,26,21,131,"Datenbank"
```

... richtet ein Fenster mit der Überschrift »Datenbank« ein und speichert den alten Inhalt.

RWINDOW(x)

... bringt die Anzahl der bisher gespeicherten Fenster. Als Parameterwert für x muß man stets die Zahl 3 angeben:

```
PRINT RWINDOW(3)
```

Hochauflösende Grafik im 80-Zeichen-Bildschirm

GRAPHIC 6, lö, anfang: ... schaltet die Hires-Grafik des VDC-Screens mit 640 x 200 Punkten ein (nur mit dem erweiterten VDC-RAM 64 KByte möglich). Die Grafik beginnt bei »anfang« x $4000 (16384). Der Parameter »anfang« darf Werte von 0 bis 3 annehmen. Ist lö = 1, wird der Bereich vorher gelöscht.

GRAPHIC 7, lö, anfang: ... aktiviert die 640 x 176-Pixelgrafik (jeweils 8 x 8-Punkte mit gleicher Vorder- und Hintergrundfarbe). Per COLOR 2,x (Hintergrund) und COLOR3,x (Vordergrund)l lassen sie sich jederzeit für den nächsten Zeichenbefehl ändern.

Die folgenden Einstellungen sind nur mit dem 64-KByte-VDC-Speicher möglich (z.B. im C-128D-Blech):

GRAPHIC 8, lö, zeile: ... 640 x 800 Punkte, wobei 640 x 200 gezeigt werden. »zeile« gibt an, wo der 640 x 200-Pixelbereich beginnt und kann zwischen 0 und 600 liegen.

GRAPHIC 9, lö, anfang: 640 x 360 Pixel. Position im Speicher: bei anfang = 0 ab $0000, anfang = 1 ab $8000.

GRAPHIC 10, lö, anfang: 640 x 400 Punkte. (anfang = 0: $0000, anfang = 1 ab $8000.

GRAPHIC 11, lö, anfang: 720 x 350 Punkte. (anfang = 0: $0000, anfang = 1 ab $8000).

Diese Basic-7.0-Anweisungen lassen sich nun wie gewohnt auch im VDC-Grafikmodus einsetzen:

BOX, CIRCLE, DRAW, GRAPHIC, PAINT, GSHAPE, SSHAPE, LO-

CATE und der neue CHAR-Befehl.

Sind beide Grafik-Modi des C128 aktiv (40 und 80 Zeichen bei zwei Monitoren), wirken die Befehle nur im 40-Zeichen-Modus.

CHAR [farb],x, y,[winkel], string$

Der CHAR-Befehl wurde erweitert und läßt sich in beiden Grafik-Modi nutzen. Bei normaler Schreibweise (ohne Parameter »winkel«) greift der Computer auf die Originalroutine zurück (nur in den beiden Textmodi und bei 40-Zeichen-Grafik anwendbar), beim neuen Befehl rangiert der VIC-Grafikmodus ebenfalls vor VDC-Grafik.

Die Zeichenkette »string$« kann mit »winkel« entgegen des Uhrzeigersinns gedreht werden (winkel = zwischen 0 und 360).

Im Textstring darf man diese Steuerzeichen benutzen:

- CHR$(17): ... setzt die nachfolgenden Zeichen vier Punkte tiefer,
- CHR$(145): ... vier Punkte höher,
- CHR$(14): Umschaltung zur Kleinschrift. Der Zeichensatz liegt in Bank 14. Das High-Byte des Speicherbereichs, in dem die Zeichenmuster beginnen, steht in $11EB.
- CHR$(142): ... stellt wieder Großschrift ein. Das High-Byte findet man dann in $11EC.
- CHR$(18): Alle nachfolgenden Zeichen erscheinen revers.
- CHR$(146): Der Revers-Modus wird abgestellt.
- CHR$(2): ... unterstreicht alle Zeichen die folgen,
- CHR$(130): ... schaltet Unterstreichen aus,
- CHR$(21): ... aktiviert komprimierte Schrift,
- CHR$(22): ... stellt komprimierte Schrift wieder ab,
- CHR$(23): Proportional-Schrift ein. Allerdings gibt's einen Schönheitsfehler: Nach acht Pixeln schmuggelt sich durch die Berechnung des Drehwinkels ein Punkt dazwischen.
- CHR$(24): Proportional-Schrift aus.

LOW(RGR(0))/ HIGH(RGR(0))

Bei Masterbasic 128 läßt sich mit dieser Funktion auch der VDC-Grafikmodus ermitteln. Ist er abgeschaltet, liefert LOW(RGR(0)) den normalen Wert. HIGH(RGR(0)) bringt 0 bzw. sechs bis elf (je nach eingestellter Grafikart.

RDOT

- RDOT(3): ... gibt die horizontale Pixelbreite der Grafik aus,
- RDOT(4): ... die vertikale Ausdehnung,

```
graphic 8: print rdot(3), rdot(4)
 640 800
```

Zusätzlich stellt Masterbasic 128 erweiterte Blockgrafik für den 40- oder 80-Zeichen-Textbildschirm zur Verfügung: Auf jeder Bildposition werden vier Punkte angezeigt. Der 40-Zeichen-Screen hat nun eine maximale Auflösung von 80x50 Punkten. Bis auf SSHAPE und GSHAPE lassen sich alle Grafikbefehle uneingeschränkt anwenden. Die Original-CHAR-Anweisung bringt das gewohnte Ergebnis, die neue liefert vergrößerten Text. Hier sollte man entweder mit komprimierter Schrift oder mit der Anweisung WIDTH 2 arbeiten! Die erweiterte Blockgrafik läßt sich nur nutzen, wenn kein anderer Grafiktyp aktiv ist, also RGR(0) = 0 bzw. RGR (0) = 5.

Achtung: Unterbrechen Sie keine Programme mit <RUN/STOP RESTORE>, die VDC- oder Blockgrafikbefehle verwenden – sonst stürzt der Computer durch die geänderte Positionierung der Common Area ab!

Geänderte Funktionen

v = SQR(x[,wurzel])

Defaultwert für »wurzel«: 2.

```
print sqr(4),sqr(15625,3)
```

DEC(numerische variable)

Die Funktion DEC fungiert auch als Befehl: Der Variableninhalt wird um »1« reduziert.

ERR$(x)

Neue Fehlernummern zwischen 42 und 47 wurden zusätzlich integriert:

- 42: label not found,
- 43: too many windows,
- 44: too many subroutines,
- 45: proc not found,
- 46: endproc without proc,
- 47: proc without endproc.

Der Wirkungskreis des ESCAPE-Key wurde komfortabel erweitert. Man drückt nach <ESC> eine der folgenden Tasten:

- <ESC> <0>: ... erzeugt einen Reset (entspricht dem Direktbefehl BANK15: SYS $FF3D),
- <ESC> <1>: ... schaltet die ASCII-/DIN-Taste ab und aktiviert den DIN-Zeichensatz,
- <ESC> <2>: Nach Deaktivierung der ASCII-/DIN-Taste wird der ASCII-Zeichensatz eingestellt,
- <ESC> <3>: Die ASCII-/DIN-Taste funkioniert wie gewohnt,
- <ESC> <4>: ... entspricht dem BOOT-Befehl (ohne Parameter),
- <ESC> <5>: Nach Anführungsstrichen (Quote-Modus) läßt sich die ESC-Taste als Grafikzeichen angeben – statt:

```
print chr$(27)"1"
```

also:

```
print "esc5 1"
```

Die ESC-Codes werden nun bei der Listing-Ausgabe auf dem Bildschirm sofort ausgeführt. Damit lassen sich die tollsten Effekte erzeugen, (Text revers, Listing-Zeilen verstecken usw.).

- <ESC> <6>: ... schaltet auf den Revers-Modus des 80-Zeichen-Schirms,
- <ESC> <7>: ... und wieder aus,
- <ESC> <8>: ... meldet den aktuellen Fehler-Status (entspricht PRINT DS$).
- <ESC> <9>: Umschalten zum C-64-Modus (wie GO 64 ohne Sicherheitsabfrage).
- <ESC> <+>: ... öffnet in der Bildschirmmitte ein Hilfs-Window (für kurze Berechnungen, Monitor-Kommandos usw.).
- <ESC> <->: ... deaktiviert das Fenster und holt den alten Bildschirminhalt zurück. Die Zeilenverknüpfungs-Tabelle wird auf die ursprünglichen Werte gesetzt.
- <ESC><HOME>: ... entspricht »Anti-HOME«: Der Cursor steht jetzt in der letzten Zeile.

Neue Anweisungen und Kommandos

Nahezu alle Parameter lassen sich auch durch Variablen oder Rechenausdrücke ersetzen. Parameter in eckigen Klammern sind optional (kann man also weglassen!).

CENTRE (str$[,zeile]) ...

... zentrierte Ausgabe der Zeichenkette str$ in der aktuellen Zeile. Steuerzeichen werden bei der Positionsberechnung nicht berücksichtigt. Die drei Punkte »...« bedeuten, daß anschließend zum Print-Befehl verzweigt wird.

Es dürfen Doppelpunkt, Komma, Semikolon, die Anweisung USING usw. folgen. Ist es das Befehlsende, setzt das Programm ein <RETURN> (= CHR$(13)). Mit dem Semikolon bleibt der Cursor hinter dem letzten Buchstaben stehen.

```
10 centre("Versuch 1",10);"in Zeile";10
20 centre(chr$(7)+"Versuch 2")
30 centre(chr$(2)+"Versuch 3"+chr$(130));:
 print"Test"
```

Bei den Ausgaben von VERSUCH 2 ertönt ein Klingelzeichen, VERSUCH 3 wird auf dem VDC-Schirm unterstrichen, der Cursor bleibt hinter der Zahl 3 stehen – »TEST« erscheint unmittelbar danach.

CENTRONIC [ga][,sekgross][,sekklein]

CENTRONIC OFF

Ein- bzw. Ausschalten der Software-Centronics-Schnittstelle.

Die Daten des Assembler-Programmteils werden ins RAM ab $0D00 kopiert. Als Geräteadresse gilt »4« bzw. ga, die Sekundäradresse für Großschrift ist 0 (sekgross), für Kleinschrift 7 (sekklein). Alle anderen Werte für die Sekundäradresse sind Linearkanäle: Die zu übertragenden Bytes werden nicht verändert. Wenn der DIN-Zeichensatz aktiv ist, werden Umlaute und Sonderzeichen angepaßt.

CENTRONICOFF schaltet wieder die Standard-Druckroutinen ein, ebenso <RUN/STOP RESTORE>.

Beispiele:

```
centronic: rem einschalten mit 4,0,7
centronic ,1: rem einschalten mit 4,1,7
centronic 5: rem einschalten mit 5,0,7
centronic off: rem abschalten
```

CGOSUB zeilennummer

»zeilennummer« darf ein beliebiger, ganzzahliger Ausdruck sein, der z.B. aus einer Rechenfunktion resultiert (berechnetes GOSUB).

Anmerkung:»zeilennummer« kann nur im Bereich 0 bis 63999 liegen. Die berechnete Zeilennummer muß existieren, sonst erscheint eine Fehlermeldung. Die aktivierte Unterroutine verläßt man wie gewohnt per RETURN-Anweisung.

```
100 cgosub x*100+1000
```

Bei X = 10 verzweigt das Programm also zu Zeile 2000, bei X = 1 zu Zeile 1100.

CGOTO zeilennummer
... wie CGOSUB, aber ohne Rücksprung in den aufrufenden Programmteil (s. GOTO des Basic 7.0).

CHANGE//suchobjekt// TO //tauschobjekt// [anf][- [end]
Der Computer tauscht »suchobjekt« gegen »tauschobjekt« aus, allerdings nur im angegebenen Zeilenbereich »anf« bis »end«. Sämtliche Leerzeichen werden berücksichtigt.

Die verschiedenen Formen des Parameters suchobjekt:
- besteht aus beliebigen Kommandos, Variablen, Buchstaben: Das Programm forscht außerhalb von String-Texten oder Zeichenketten,
- beginnt und endet mit Anführungsstrichen: Jetzt sucht der Computer die beiden Begriffe nur in Anführungszeichen gekleideten Text-Strings.

Findet der Computer das Suchobjekt, erscheint die Zeile auf dem Bildschirm und die Frage, ob man tauschen, weitersuchen oder abbrechen will. Ist das Tauschobjekt leer (////), wird das Suchobjekt gelöscht. Wenn's die einzige Anweisung ist, wird die gefundene Zeile komplett gelöscht.

```
change //print"a"// to //print"egal"//
```

Damit ersetzt man im gesamten Basic-Programm alle Anweisungen PRINT »A« durch PRINT »EGAL«.

```
change //end// to //// 10 - 60
```

... löscht das Kommando »end« in den Zeilen 10 bis 60.

```
change //"dummy"// to //"text"// - 1000
```

Nur in Zeichenketten (Strings) der Zeilen bis 1000 wird das Wort »dummy« durch »text« ersetzt.

DISK kommando$[,ga]
... sendet den Floppybefehl »kommando$« über den Befehlskanal 15 an die Diskettenstation ga. Voreingestellt ist 8.

```
disk "v"
```

... validiert die Diskette in der Floppystation mit Gerätenummer 8.

DMERGE datei[,u ga][, d drive]
... fügt ans Programm im Speicher das andere Basic-Programm »datei« an (von Gerät ga und Drive drive).

Anmerkung: Parameter siehe DLOAD. Bei Angabe von Variablen müssen diese in Klammern stehen. Das Programm »datei« muß höhere Zeilennummern besitzen als das im Speicher.

```
dmerge "datei",u 8
dmerge (n$), u(g), d(d%)
```

DOKE adresse,wert
... entspricht einem Doppel-POKE. Das Low-Byte von »wert« wird in »adresse« gespeichert, das High-Byte in »adresse + 1«.

»adresse« und »wert« können Werte zwischen 0 und 65 535 annehmen. Es gilt immer die aktuelle Bank (eingestellt per Anweisung BANKX).

```
doke dec("2f"),dec("2000"):clr
```

... setzt den Variablenanfang auf $2000.

DUMP
... listet alle nichtindizierten Variablen und definierte Funktionen.

```
10 deffn f(x)=sin(x)
20 a$="string"
30 a%=1
40 dump
run
fn f(x)=sin(x)
x= 0
a$="string"
a%= 1
```

ESCAPE
... gibt eine Kurzübersicht der ESC-Funktionen aus.

Auf der linken Seite erscheint das Zeichen, dessen Taste nach ESC zu drücken ist.

FILL bank, von, bis, code
... füllt den Bereich »von« bis »bis« in Bank »bank« mit dem Wert »code«.

»bank« kann Werte von 0 bis 16 annehmen (16 = VDC-RAM). »von« ist die Startadresse, »bis« die letzte zu füllende Speicherstelle. »code« liegt zwischen 0 und 255.

Beispiel:

```
fill16,0,4095,0
```

... schreibt ein Null-Byte in den VDC-Textspeicher und ins Attribut-RAM.

```
fill0,1024,2023,32
```

... löscht den 40-Zeichen-Bildschirm. Das Farb-RAM bleibt erhalten.

FIND//objekt// [von][-][bis]]
... listet alle Zeilen des definierten Bereichs mit »objekt«.

Sucht man eine Zeichenfolge, die irgendwo in einem String steht, müssen an Beginn und Ende Anführungszeichen gesetzt werden (z.B. FIND //"text"//).

Ist das erste Zeichen nach // kein Anführungszeichen, werden bei der Suche alle String-Texte im Programm überlesen. Anderfalls berücksichtigt die Befehlsroutine nur die String-Texte.

Leerzeichen (Spaces) in »objekt« werden nicht übergangen und müssen auch im Programm an der gleichen Stelle stehen – sonst kann sie die Routine nicht finden.

Durch die Codenummern (Basic-Token) der neuen Befehle werden auch Zeilen geLISTet, die das gesuchte Wort garnicht enthalten. Beispiel: die Anweisung END. END besitzt den Token-Wert $80, die Anweisung DOKE hat die Nummern $FE $80. Bei der Suche nach END entdeckt die Routine auch DOKE und bringt zusätzlich alle Basic-Zeilen, die diesen Befehl enthalten.

```
10 print"Test der FIND-Anweisung"
20 printa,f%
30 dim ab(10,10)
40 goto 20
```

Damit sind folgende Abfragen möglich:

```
find //20//': rem listet Zeile 40.
find //"find"// - 30: rem bringt Zeile 10
find //a//: rem Zeilen 20 und 30
open 1,4:cmd1:find //a// 10 -25:print1:close1:
rem nur Zeile 20 zum Drucker
```

GCOPY getmode, x1, y1, x2, y2, putmode, x, y
... verschiebt Grafikausschnitte – auf dem aktuellen Screen oder zwischen verschiedenen Grafikbildschirmen.

Für »getmode« und »putmode« gelten die Modi des GRAPHIC-Befehls: Die Modi 1 bis 4 und 6 bis 11 müssen aktiv sein (aktivierte Hires-Grafik auf dem Screen), also:

```
graphic getmode:graphic putmode: gcopy...
```

Wurde als Modus 0 oder 5 gewählt, ist die erweiterte Blockgrafik aktiv (s. Beschreibung). Die Eckpunkt-Werte dürfen keine Fantasiezahlen sein: Die Koordinaten x1/y1 werden nach x/y übertragen, x2/y2 sind dann die rechte untere Ecke der neuen Grafik. Damit haben Sie die Möglichkeit, Grafiken entweder um die x- bzw. y-Achse oder beide zu spiegeln. Diese Werte kann man einstellen:
- x1 < = x2 : x-Achse normal, x1 > x2 : x-Achse gespiegelt
- y1 < = y2 : y-Achse normal, y1 > y2 : y-Achse gespiegelt

Einziger Nachteil: Nur die Pixel-Information wird verschoben, nicht aber das Attribut-RAM (Farben usw.).

GLOBAL
... stellt den Urzustand der Variablen vor der Anweisung LOCAL (s. dort) wieder her.

Bemerkung: LOCAL und GLOBAL lassen sich auch schachteln.

Wird GLOBAL ohne vorhergehendes LOCAL gesetzt, schützt der Anfang der Common-Area vor Absturz: Der Befehl hat keine nachteiligen Wirkungen. Wurde der Variablenanfang (vor LOCAL) auf eine andere Adresse gesetzt, z.B. auf $1250 statt $1000 (Anfang der Common-Area), erzeugt GLOBAL im C128 ein Chaos: Der Computer versucht nämlich aus den zufälligen Werten vor $1250 gültige Variablengrenzen zu konstruieren. Die nach LOCAL definierten Variablen sind dann natürlich gelöscht.

INC (variable)
... erhöht den Inhalt der Variablen um 1.

Bemerkung: Ist es eine Zeichenkette, wird der String auf einen Buchstaben reduziert und dessen ASCII-Wert erhöht. CHR$(255) bringt dann z.B. CHR$(0) als Ergebnis. Für einen Leerstring setzt der C128 CHR$(0) und gibt CHR$(1) zurück.

INFO kommando
... gibt zum Befehl »kommando« eine kurze Beschreibung.

Bemerkung: Funktionen werden als Zuweisung an die Variable »v« oder »v$« angezeigt, ohne v oder v$ zu verändern.

LOOP, UNTIL und WHILE lassen sich nicht abfragen, da diese eng mit DO verknüpft sind, ebenso STEP (s. FOR).

Beispiel:

```
info dec
v=dec(hex$)
dec(var)
```

INPUT* lf, var$, länge

... holt über den Datenkanal »lf« Zeichen bis zur Anzahl »länge«. Fehlt dieser Parameter, holt der Befehl so lange Bytes in den Computer, bis der Code CHR$(13) (= RETURN) auftaucht.

»var$« ist eine beliebige Stringvariable. Die Anweisung akzeptiert jedes Zeichen – egal ob Komma, Anführungszeichen, CHR$(0) usw. Ist »länge« größer als die Anzahl der gespeicherten Zeichen, füllt die Routine automatisch mit CHR$(13) auf.

Beispiel:

```
10 dopen#1,"Test",W
20 print#1,"Dies ist ein Versuch"
30 dclose#1
40 dopen#1,"Test"
50 input*1,a$
60 dclose#1
70 dopen#1,"Test"
80 input*1,b$,6
90 dclose#1
100 printa$:printb$
run
Dies ist ein Versuch
Dies i
```

INPUT> in$, x, y, lä, vorgabe$, end$, erlaubt$

... erlaubt formatiertes Input mit individuell definierbaren Einschränkungen. Erläuterung zu den Parametern:
- in$: beliebige Stringvariable,
- x: Anfangsspalte des Input-Feldes (die aktuelle Position gilt als Vorgabe).
- y: Zeile des Input-Feldes,
- lä: Länge des Eingabefeldes. Vorgegeben ist die Entfernung von »x« bis zur rechten Bildschirmgrenze – je nach Fensterdefinition, also maximal 80 Zeichen.
- vorgabe$: ... wird vorrangig im Input-Feld ausgegeben und darf nicht größer als »lä« sein. Fehlt vorgabe$, löscht der Computer das Input-Feld.
- end$: Damit lassen sich (außer <RETURN>) noch maximal 50 andere Tasten definieren, die die Eingabe abbrechen (z.B. <CRSR abwärts>).
- erlaubt$: ... legt die bei der Eingabe zugelassenen Tasten fest, z.B. »jJnN«. Als Vorgabe dienen alle Tasten mit den ASCII-Codes 32 bis 127 und 160 bis 255.

Bei aktiviertem Eingabefeld besitzen folgende Tasten Sonderfunktionen:
- <ESC>: ... löscht das Input-Feld und holt die Vorgabe zurück,
- <CLR/HOME>: ... macht das Eingabefeld frei,
- <HOME>: ... bewegt den Cursor zur ersten Position,
- <CRSRlinks/rechts>: Cursor-Bewegung im Input-Feld,
- <DEL>: Zeichen vor dem Cursor löschen und restlichen Text nachziehen,
- <INST/DEL>: Die Zeichen werden ab Cursor-Position um eins nach rechts verschoben und ein Leerzeichen eingefügt.

Nach Eingabenende (per Abbruchtaste) überträgt der Computer den Inhalt des Input-Felds in die Variable »in$« (ohne nachfolgende Leerzeichen). Die Nummer der Abbruchtaste steht in der Variablen »st«: 0 ist z.B. <RETURN>. Ist »st« größer als 0, bezeichnet der Wert die Zeichenposition in »end$«.

Wer versucht, nicht gestattete Zeichen einzutragen, provoziert das Warnsignal CHR$(7) – ebenso, wenn man den rechten Rand überschreiten will.

Beispiel:

```
input>a$,0,0,,"Testeingabe",chr$($11)+chr$
($91),"abcdefghijklmnopqrstuvwxyz"
```

Das Input-Feld erstreckt sich ab der linken oberen Bildschirmecke bis zum eingestellten Rand. Ab Position 0,0 erscheint »esteingabe« im Feld (ohne »T«, da dieser Großbuchstabe nicht im Parameter erlaubt$ vorkommt). Die Eingabe läßt sich mit <RETURN> oder <CRSR-abwärts/aufwärts> beenden. Die Eingabe steht jetzt in A$, die Nummer der Abbruchtaste in der Systemvariablen st.

KILL var[,arr(0..0)[,...]]

... löscht die angegebenen Variablen oder Arrays (Variablenfelder).

Man darf beliebig viele Variablen und Arrays angeben. Sie werden aus der Liste gestrichen, der Rest rückt nach.

LABEL labelname

... dient dazu, eine Programmzeile mit einem Namen zu kennzeichnen (labelname).

Das Label wird nur dann erkannt, wenn's am Zeilenanfang (unmittelbar hinter der Zeilennummer) steht. Lediglich Doppelpunkte oder Leerzeichen kann man noch davor einfügen.

Achtung: Der Labelname darf keine Doppelpunkte enthalten, ebenso sollten Anführungszeichen entfallen oder zumindest paarweise benutzt werden; sonst werden nachfolgende Befehle nicht erkannt.

Zwischen LABEL und labelname können jedoch beliebig viele Leerzeichen stehen.

Zum Labelnamen gehören aber Leerzeichen unmittelbar vor dem Befehlsende:

```
10 label test :
```

ist etwas anderes als

```
10labeltest:
```

Nach »labelname« kann man das Programm fortsetzen, wenn der nächste Befehl durch einen Doppelpunkt abgetrennt wurde:

```
1000 : label print: print"LABEL print erreicht."
```

LGOSUB labelname

... quasi ein GOSUB-Sprung zur markierten Zeile »labelname«.

Falls der Label nicht existiert, erscheint der Fehler »LABELNOTFOUND«. Per <RETURN> verläßt man solche Label-Subroutinen.

```
10 lgosub print
20 print "Zurück im Hauptprogramm"
30 end
100 label print
110 print"LABEL print erreicht."
120 return
```

LGOTO labelname

Sprung zur markierten Zeile »labelname«.

LLIST [labelname]

... bringt das Programm-Listing ab Zeile »labelname« auf den Screen. Ohne Label-Angabe erscheinen alle Zeilen, die einen Label-Namen enthalten.

LOCAL

Nach der Eingabe von LOCAL herrscht im Computer der gleiche Zustand wie nach CLR (Variablenspeicher löschen). Aber: Vorher definierte Variablen werden nicht gelöscht, lediglich die alten Variablenzeiger gespeichert und die neuen auf den noch verbleibenden Restspeicher gesetzt.

LRESTORE labelname

... setzt den Lesezeiger der READ-Anweisung aufs erste Datenelement nach der Zeile »labelname«.

Beispiel:

```
10  lrestore data2:read a,b,c
20  lrestore data1:read d,e,f
30  print a;b;c;d;e;f
100 label data1:data 10,20,30
200 label data2
210 data 1,2,3
run
1 2 3 10 20 30
```

MATRIX [arr(0..0)[,arr(0..0)[,...]]]

... listet Arrays und gibt deren Werte aus.

MAXINP in$(i1), x(i2), y(i3), l(i4), vorgabe$(i5), ende$, erlaubt$(i6), anzahl

... erzeugt Bildschirmmasken, in der man alle Eingaben in einem Rutsch erledigen kann.

MAXINP definiert »anzahl« Eingabefenster und schreibt die jeweilige Vorgabe ins entsprechende Feld.
- in$(i1): Dieses Variablenfeld enthält die Eingaben ab DIM i1,
- x(i2): x-Koordinate des Eingabefensters,
- y(i3): y-Koordinate,
- l(i4): Breite des Eingabe-Windows. Bei Länge = 0 erstreckt sich das Fenster bis zum Bildschirmrand.

– vorgabe$(i5): eventuell definierte Zeichenkette als Vorgabe, kann auch leer sein.
– ende$: ... enthält alle Tasten-Codes, die zwischen den Fenstern umschalten oder die Eingabe beenden. Das erste Zeichen führt ins nächste Fenster, das zweite Zeichen ins vorhergehende; alle weiteren provozieren das Eingabenende.
– erlaubt$(i6): ... bezeichnet Tastatur-Codes, die man im Eingabefeld benutzen kann. Ist ein Array-Element leer, sind alle Tasten der ASCII-Codes 32 bis 127 und 160 bis 255 erlaubt.
– anzahl: Menge der Eingabefenster.

Alle Array-Elemente, die auf derselben Ebene liegen, gehören zusammen: z.B. definieren bei MAXINP a$(1), x(10), y(4), ... die Variablen a$(1), x(10), y(4), ... ; a$(2), x(11), y(5), ... jeweils eine Ebene (zur Benutzung der einzelnen Eingabefenster: s. Beschreibung zum INPUT>-Befehl).

Achtung: Die Basic-Erweiterung testet nicht, ob die Arrays entsprechend groß dimensioniert wurden! Bei einer Überschreitung der DIM-Grenzen gerät die gesamte Variablenverwaltung aus den Fugen.

Für x, y und l lassen sich sowohl Real-(Fließkomma-) als auch Integer-Arrays verwenden.

Beispiel:

```
10  dim a$(3,2),x(3),y(3),l%(3),v$(3),e$(3)
20  x(1)=5: y(1)=3: l%(1)=20: v$(1)="Name"
30  e$(1)="abcdefghijklmnopqrstuvwxyz"
40  x(2)=5: y(2)=5: l%(2)=0: v$(2)="Adresse"
50  e$(2)=""
60  x(3)=5: y(3)=7: l%(3)=10: v$(3)="Telefon
    nummer"
70  e$(3)=""
80  scnclr: en$= chr$(17)+chr$(145)+chr$(13)+"*"
90  maxinp a$(1,1),x(1),y(1),l%(1),v$(1),en$,e$
    (1),3
100 maxinp a$(1,2),x(1),y(1),l%(1),v$(1),en$,e$
    (1),3
```

Zunächst definiert man die Arrays:
– Fenster 1: Pos. 5, 3, Länge 20, Vorgabe»ame« (»N« ist nicht erlaubt), Kleinbuchstaben gestattet,
– Fenster 2: Pos. 5, 5, Länge max., Vorgabe»Adresse«, alle Tasten sind erlaubt,
– Fenster 3: Pos. 5, 7, Länge 10, Vorgabe»Telefonnum«, alle Tasten gestattet.

Zwischen den einzelnen Windows kann mit <CRSR aufwärts/abwärts> wechseln. Per <RETURN> oder <*> verläßt man alle Fenster. Die drei Eingaben ab a$(1,1) und a$(1,2) werden gespeichert.

MEMORY

... zeigt die aktuelle Speicherbelegung.

Zu den Optionen »Programm«, »Speicher frei in Bank 0«, »Variablen«, »Arrays«, »Speicher frei in Bank 1« und »Stringspeicher« erhält man Anfangs- und Endadresse als Hexadezimalzahl; den jeweiligen Umfang des Speicherbereichs in dezimaler Form.

MERGE [datei[,ga]]

... fügt ans Programm im Speicher ein anderes (»datei«) vom Gerät »ga« an.

OLD

... erweckt ein per Reset oder NEW gelöschtes Basic-Programm wieder zum Leben.

Achten Sie nach einem Reset darauf, ob vorher ein Grafikbereich aktiviert war. Falls ja, sollte man folgende Befehlszeile im Direktmodus absetzen:

```
graphic 1: graphic 0: old
```

PCLOSE

Unterbricht man PPRINT oder PLIST (s. Beschreibung) oder tritt bei der Datenübertragung ein Fehler auf, läßt sich per PCLOSE der CLOSE IF-Befehl quasi nachholen.

PCOPY [mode[,[lf][,[ga][,sec]]]]

... schickt den Inhalt eines Grafik- oder Textbildschirms zum Drucker.

Der Defaultwert von »mode« ist der Modus, der als erster lt. folgender Liste zutrifft:
– 0: 40-Zeichen-Grafik,
– 1: VDC-Grafik,
– 2: 40-Zeichen-Text,
– 3: 80-Zeichen-Text.

Die anderen Parameter: s. OPEN-Befehl (Voreinstellung: lf=1, ga=4, sec=1.).

Beim VDC-Grafik-Druck werden keine Interlace-Bilder unterstützt. Im 80-Zeichen-Textbildschirm dürfen auch andere Formate aktiv sein, z.B. 82 Zeichen zu je 20 Zeilen oder 45 Zeichen mit 31 Zeilen.

Für alle Hardcopies wird stets der Grafikmodus des Druckers eingeschaltet. Der 40-Zeichen-Textschirm holt die Zeichendaten aus BANK 14, im 80-Zeichen-Modus kommen die Muster aus dem VDC-RAM. Die im entsprechenden Attribut-RAM eingestellten Optionen wie Unterstreichen, Revers oder Zeichensatz-Umschaltung werden berücksichtigt.

Die Druckroutine aktiviert stets die höchste Auflösung des Geräts (8-Nadel-Drucker: ESC Z, 24-Nadler: ESC * (40)). Ist der Bildschirm schmaler als 81 Zeichen, werden alle Grafik-Bytes dreimal hintereinander gedruckt. Sind's mehr als 80 Zeichen, druckt die Routine nur zweimal – das Bild ist dann verzerrt. 24-Nadler produzieren jeweils einen zusätzlichen Druckdurchgang.

Beispiele:

```
pcopy ,4,5
```

... druckt den aktuellen Screen mit Gerät 5 und logischer File-Nummer 4.

```
pcopy 1,,,0
```

... VDC-Grafik zu Gerät 4 mit logischer File-Nummer 1 und Sekundäradresse 0.

PLIST [von[-[bis]]

... schickt einen Auszug des Basic-Listings zum Drucker (entspricht OPEN lf, ga, sec: CMD lf: LIST von - bis: PRINT lf: CLOSE lf). Parameter zu OPEN: s. PSET.

PPRINT [ausdrücke]

... wirkt wie OPEN lf, ga, sec: PRINT lf, ausdrücke: CLOSE lf.

Es lassen sich alle Ausdrücke oder Zeichen wie bei PRINT angeben.

PSET [lf][,[ga][,sec]]

... setzt die OPEN-Parameter zu den Befehlen PPRINT, PLIST, PLCOSE. Nach dem Start von Masterbasic 128 sind die Werte »1, 4, 7« vorgegeben.

SCNLOAD datei[,ga]

... lädt ein Bildschirmfenster samt Inhalt.

Der gespeicherte Window-Bereich wird inkl. Farb- und Attribut-RAM geladen. War das gesicherte Fenster größer als das aktuell aktivierte, ignoriert die Erweiterung überzählige Spalten oder Zeilen. Ist der neu definierte Bereich größer, erscheint das Window nur maximal im gespeicherten Umfang – die Ränder des neuen Fenster bleiben leer.

Es ist egal, ob man ein 40- oder 80-Zeichen-Window lädt: Die Farben werden entsprechend angepaßt. Die voreingestellte Geräteadresse »ga« ist 8.

SCNSAVE datei[,ga]

... sichert ein Bildschirmfenster mit allen Werten des Farb- und Attribut-RAM auf Diskette (Defaultwert für ga: 8).

SCROLL u xlo,ylo,xru,yru[,[put$][,get$]]
SCROLL d xlo,ylo,xru,yru[,[put$][,get$]]
SCROLL r xlo,ylo,xru,yru[,[put$][,get$]]
SCROLL l xlo,ylo,xru,yru[,[put$][,get$]]

... verschiebt den vorgesehenen Bildschirmbereich mit den Koordinaten »xlo, ylo« (links oben) und »xru,yru« (rechts unten, bezogen aufs aktuelle Bildschirmfenster) in Richtung der entsprechenden Buchstaben:
– u: hoch,
– d: abwärts,
– r: rechts,
– l: links.

Bei Angabe der Zeichenkette »put$« erscheint dieser in der Zeile, die frei geworden ist. Steuerzeichen (ASCII-Codes 0 bis 31 und 128 bis 159) werden ignoriert.

Fügt man die Variable »get$« an, überträgt der Computer den Inhalt der weggescrollten Zeile in diesen String.

Wichtig: Bei SCROLL wird die Zeilenverknüpfungstabelle gelöscht. Resultat: BASIC-Zeilen, die über eine Bildschirmzeile hinaus in die darunterliegende ragen, gehören jetzt für den Bildschirm-Editor nicht

mehr zusammen. Will man solche Programmzeilen ändern, muß man sie vorher erneut LISTen.

SORT [Richtung],Arr([1.dim,]anf),Arr([1.dim,]end)

... sortiert den angegebenen Array-Bereich beliebigen Typs in den Grenzen »anf« bis »end«:
- Richtung = 0: aufsteigend,
- Richtung = 1: von höchsten Wert bis zum niedrigsten.

Bei eindimensionalen Arrays wird normal sortiert; zweidimensionale Felder vergleicht die Routine nach der ersten Dimension (sie muß bei beiden Angaben gleich sein). Sind's mehr als zwei Dimensionen, entfällt die Sortierung. SORT arbeitet z.B. ideal mit MAXINP zusammen.

Beispiele:

```
10 dim a%(55),p$(4,100),r(4,3,5)
sort ,a%(4),a%(20)
```

... sortiert das Array a% in den Grenzen von 4 bis 20,

```
sort ,p$(2,1),p$(1,100)
```

... erzeugt eine Fehlermeldung, da sich die ersten Dimensionen unterscheiden,

```
sort ,p$(2,1),p$(2,100)
```

... ordnet das Feld p$ in den Grenzen von 1 bis 100. Als Sortierkriterium dient die zweite Dimension (also z.B. Namen, Städte usw.)

```
sort ,r(1,1,0),r(1,1,5)
```

... provoziert ebenfalls eine Fehlermeldung, da mehr als zwei Dimensionen existieren.

STRSAV... anfangsadresse, len, var$

... liest eine Zeichenkette aus dem Speicher,

Wird ein String gelesen, müssen die zu holenden Zeichen bei »anfangsadresse« beginnen. Die Zeichenkette wird dann bis zur Länge »len« gelesen und der Variablen »var$« zugeteilt.

STRSAV str$, anfangsadresse

... schreibt den String in den entsprechenden Bereich.

Soll eine Zeichenkette gespeichert werden, wird er ab Speicherstelle »anfangsadresse« in voller Länge abgelegt.

Beide Befehlsversionen beziehen sich auf die Speicherbank, die mit BANK n eingestellt wurde.

Beispiele:

```
bank 0: strsav"Stringspeichern", $1b00
```

... überträgt die Zeichenkette in den Bereich ab $1B00 bis $1B0F,

```
bank 0: strsav$1b00,16,a$
```

... holt den String wieder aus dem Speicher und weist ihn der Variablen a$ zu.

SWAPARR arr1a(0..0),arr1b(0..0),[arr2a(0..0),arr2b(0..0)[,...]]

... vertauscht die Inhalte zweier Arrays, die gleichen Typs sein müssen.

(0..0) weist darauf hin, daß man einen beliebigen, aber gültigen Index angeben muß (die Nullenanzahl je nach Dimension).

Da nur die Variablennamen getauscht werden, können das auch unterschiedlich lange Felder sein: also Vorsicht bei der Verwendung dieses Befehls!

SWAPVAR var1a,var1b[,var2a,var2b[,...]]

... tauscht den Inhalt zweier Variablen aus. Sie müssen vom gleichen Typ sein. Es lassen sich auch Array-Elemente angeben. Bis zur Länge einer Eingabezeile kann man beliebig viele Variablen-Paare eintragen.

```
a$="Var a$":b$="Var b$"
swapvar a$,b$
print a$,b$
```

TRANS bankload,von,bis,[banksave],nach

... verschiebt einen Speicherbereich (von, bis) an eine neue Adresse (nach).

Der Parameter »bankload« bestimmt die Bank, aus der die Bytes geholt werden:
- 0 bis 15: sind die normalen Banknummern,
- 16: ... das VDC-RAM.
- von: Startadresse des zu übertragenden Speicherbereichs,
- bis: ... das letzte zu kopierende Byte.
- banksave: Zielbank (voreingestellt ist der Wert von »bankload«).
- nach: Startadresse des Zielspeicherbereichs.

Beispiel:

```
trans 0,1024,2023,1,50000
```

... kopiert den 40-Zeichen-Bildschirminhalt ab Adresse 50000 in Bank 1.

UHR [time$][,[color][,x,y]]

Damit läßt sich die Echtzeituhr des Computers einstellen und permanent auf dem Bildschirm zeigen (nicht im jeweiligen Grafik-Modus):
- time$: ... muß exakt sechs Zeichen lang sein (Struktur »HHMMSS«),
- color: ... bestimmt die Farbanzeige (Werte zwischen 1 und 16, Voreinstellung: 8 = gelb),
- x, y: ... legen die Uhrenposition auf dem Bildschirm fest. Die Werte beziehen sich stets auf den gesamten Screen, nicht aufs aktuelle Window. Im 40-Zeichen-Modus liegt die Zeichenbreite mit 40 fest, die Zeilenzahl bei 25. Der 80-Zeichen-Bildschirm benutzt die in den VDC-Registern festgelegten Werte. Damit lassen sich z.B. beim VDC-Textschirm Ausgaben in Extrazeilen realisieren.

Die maximale Breite muß um die Zahl 8 verringert werden. Beim 40-Zeichen-Textscreen sind also nur x-Koordinaten zwischen 0 und 32 zulässig.

Für beide Bildschirme (40 und 80 Zeichen) existieren eigene Register; die Uhr ist also sofort nach dem Umschalten in den anderen Bildschirmbereich an der entsprechend definierten Position sichtbar. Die Uhr erscheint in der rechten unteren Bildschirmecke.

Die Anweisung UHR ohne Parameter schaltet die Anzeige aus bzw. ein. Intern läuft die Uhr aber stets weiter, bis Sie den Computer ausschalten.

```
uhr "115500"
```

Die Uhr wird 5 Minuten vor 12 gestartet und an den zuletzt aktuellen Cursor-Koordinaten gezeigt.

```
uhr ,1,32,0
```

... zeigt die Uhrzeit im 40-Zeichen-Modus in Schwarz rechts oben, beim 80-Zeichen-Bildschirm etwa in der Mitte der ersten Zeile.

VLOAD dateiname[,d dr][,u ga][,p Adr]

... holt eine Datei ins VDC-RAM.

Es gelten die vom BLOAD-Befehl gewohnten Parameter. Trägt man dafür Variable ein, müssen man sie in Klammern setzen.

```
vload"datei",u 9
vload (n$), u(g), d(d), p(a)RL-05
```

VPOKE adresse, wert

... schreibt einen Wert ins VDC-RAM:
- adresse: ... kann je nach Größe des VDC-RAM Werte von 0 bis 16383 (VDC-Chip 8563) bzw. 0 bis 65535 (VDC-Chip 8568) annehmen.
- wert: ... darf zwischen 0 und 255 liegen.

```
vpoke 0,1: vpoke 2048,130
```

... gibt das kleine a links oben auf dem 80-Zeichen-Schirm aus.

VRPOKE reg, wert

... überträgt das Byte »wert« ins VDC-Register »reg«.

Achtung: Die Routine überprüft nicht den Wert von reg! Zahlen über »36« werden zwar akzeptiert, bringen aber kein vernünftiges Ergebnis. Die Zahl für »wert« darf nur zwischen 0 und 255 liegen.

```
vrpoke 6,20
```

... reduziert z.B. die Zeilenzahl des VDC auf 20.

VSAVE datei[,d dr][,u ga][,p Anf to p End]

... speichert einen Ausschnitts des VDC-RAM.

Es gelten die gleichen Parameter wie bei BSAVE (Variablen muß man ebenfalls umklammern):

```
vsave (n$), u(g), d(d), p(an) to p(en)
vsave "datei",d 1
```

XDEF FN func(var) = funktion$

... definiert eine Funktion (wie der Basic-7.0-Befehl DEF), allerdings muß hier eine Zeichenkette stehen.

Die Funktion FN erwartet den Funktionstext in Bank 0: Der codierte Text des Strings »funktion$« hängt sich klammheimlich hinters Basic-Programm, der Zeiger auf Programmende wird erhöht. Damit sich Ihr Basic-Listing nicht unsichtbar, aber beständig vergrößert, sollte in der ersten Programmzeile die Anweisung »OLD: CLR« stehen. Geben Sie diese Befehle auch vorsichtshalber vor jedem Speichern eines Listings ein, wenn es zuvor schon gestartet wurde.

```
10 old: clr
20 input"f(x)=";f$
30 xdef fnf(x)=f$
40 print fnf(2),fnf(4)
```

Prozeduren

Mit Masterbasic 128 lassen sich z.B. wie bei Pascal oder beim BBC

Basic V des AcornArchimedes Prozeduren definieren und aufrufen.

Was sind Prozeduren? Sie dienen dazu, einen Programmabschnitt mit einen Namen zu kennzeichnen (quasi als Label) und so das Hauptprogramm durchschaubarer zu machen. Prozeduren sind separate Programmabschnitte, die keine Rücksicht auf bereits verwendete Variablennamen nehmen müssen: Man kann sie als LOCAL definieren. Prozeduren können zwar Variablen des Hauptprogramms lesen, sind aber nicht von diesen Werten abhängig. Lokale Variablen werden bei Aufruf der Prozedur eingetragen und beim Rücksprung ins Hauptprogramm wieder eliminiert – man kann also keine Werte zwischenspeichern.

Die Adressen für den Rücksprung legt Masterbasic 128 im freien Speicher der Bank1 ab ($0400 bis $0FFF) und bietet Platz für maximal 768 Aufrufe. Sind's mehr Prozeduren, entsteht die Fehlermeldung »TOOMANYSUBROUTINES«.

Wurde eine Prozedur aktiviert, darf die computerinterne Suche nach dem entsprechenden Namen und die Parameterauswertung nicht unterbrochen werden – weder per <RUNSTOP>, <RESTORE> oder durch die Kombination beider Tasten – sonst stürzt der C128 ab.

Falls ein Programmfehler laufende Prozeduren unterbricht, sollte man versuchen, durch den Direktbefehl ENDPROC wieder ins Hauptprogramm zurückzukehren (funktioniert in 95 Prozent aller Fälle!).

So definiert man Prozeduren:

```
PROC name[(var1[,...[,& vari]...])]
...
Prozedurprogramm
...
ENDPROC [name]
```

(Der Prozedurenname nach ENDPROC kann auch entfallen, da Anweisungen in der restlichen Zeile nicht mehr ausgeführt werden.)

Aufruf einer Prozedur (im Programm oder im Direktmodus):

```
'name[(ausdruck1[,...[,vari]...])]'
```

Achtung: Prozedurname und Parameterliste muß man zwischen Hochkommas <SHIFT 7> setzen!

Die Anweisungen PROC und ENDPROC dürfen nur am jeweiligen Zeilenanfang stehen und lassen sich nur per Doppelpunkt oder Leerzeichen einrücken.

Alle Variablen, die als aufrufende Parameter erscheinen, müssen bereits vorher im Hauptprogramm definiert worden sein.

Nach dem Prozeduraufruf werden alle Zeilen zwischen PROC und ENDPROC überlesen: Prozeduren darf man also an beliebiger Stelle im Programm definieren, aktivieren sollte man sie nur per Namen. Sprünge zur entsprechenden Zeilennummer (mit GOTO) sind aufgrund der Speicherverwaltung des Basic-Interpreters zwar möglich – aber lassen Sie's besser: Es widerspricht allen Regeln strukturierter Programmierung und führt früher oder später nur zur Verwirrung.

Prozeduren lassen sich auch schachteln, aber beachten Sie:

Die Suche nach einer Prozedur läuft ebenfalls geschachtelt ab, d.h., wurde bereits eine Prozedur aufgerufen, haben die dort eingetragenen Prozeduren Vorrang vor den extern definierten Prozedur-Routinen.

Um die Ablaufgeschwindigkeit des Programms zu steigern, sollte man alle Prozeduren in den ersten Zeilen des Hauptprogramms unterbringen, ebenso sollte man Unterprozeduren direkt nach der Hauptprozedur plazieren:

```
100 proc p1
110 :  proc p2
120 :  ..m1
130 :  endproc p2
140 ..m2
150 : proc p3
160 : ..m3
170 :    proc p4
180 :     proc p5
190 :     ..m7
200 :     endproc p5
210 :   ..m4
220 :   endproc p4
230 : endproc p3
240 : ..m5
250 endproc p1
260 ..m6
```

Unsere Tabelle zeigt die unterschiedlichen Prioritätsstufen von Prozeduren.

Damit eine Prozedur auf Variablen des Hauptprogramms reagiert, übergibt man sie als Parameter, aber: Sie müssen unbedingt vorher definiert werden! Bei der Übergabe gibt's zwei Variationen:

– Wertübergabe: Beim Aufruf darf das ein beliebiger Ausdruck sein. Verläßt man die Prozedur wieder, bleiben die vor dem Aufruf definierten Variablen unverändert:

```
proc wert(a,b$,q%)
...
endproc
'wert(x,c$+left$(w$),12)'
```

Achtung: Die Variablen x, c$ und w$ muß man schon vor dem Prozedur-Aufruf einrichten!

– Namensübergabe: Hier sollte beim Aufruf eine definierte Variable stehen, deren Wert der lokalen übergeben wird. Das darf auch ein Array-Element sein. Verläßt man die Prozedur, überträgt die Erweiterung deren aktuellen Wert in die beim Aufruf gültige Variable. Die korrespondierenden Variablen müssen nicht gleichnamig sein:

```
proc name(&a$, &c%)
...
endproc
'name(w$(4,m),p%)'
```

Auch hier gilt: m, w$(1.dim,2.dim) und p% müssen bereits als gültige Variablen vorhanden sein.

Die beiden Übergabemöglichkeiten lassen sich auch mischen. Falls einzelne Variablen nicht ausreichen, kann man auch ganze Arrays übergeben. Dabei darf als Feld allerdings nur »arr(0)« mit Dimension und Länge 0 im Prozedurkopf eingetragen werden. Im Aufruf muß ein Array gleichen Typs stehen (String, Real- oder Integerzahl); die Angabe des Array-Elements ist beliebig – am besten gibt man stets (0..0) an. Das Array in der Prozedur erhält jetzt durchs Feld im Aufruf die Dimensionen-Anzahl und deren Länge.

Arrays lassen sich entweder per Wert- oder Namensübergabe weiterverarbeiten:

```
proc array(&a$(0), w(0))
...
endproc
dim q$(4,4,3), p(1000)
'array(q$(0,0,0),p(0))'
```

Das Array a$ bekommt in der Prozedur jetzt automatisch die Dimension von q$, also a$(4,4,3), w erhält die Dimension w(1000). Beide Arrays enthalten selbstverständlich nicht nur die Dimension, sondern auch den Inhalt der korrenspondierenden Arrays. Durch ENDPROC wird der eventuell geänderte Inhalt von a$ in die Variable q$ des Hauptprogramms zurückkopiert. p bleibt unverändert, da kein Name übergeben wurde.

Diese Prozedur sichert einen Speicherbereich auf Diskette:

```
10 proc speichern(n$,g,v,b)
20 print"Das File";n$;"wird von";v;
30 print"bis";b;" gespeichert."
40 bsave(n$),u(g),p(v) to p(b)
50 endproc speichern
```

Diese Prozedur zeigt den aktuellen Disk-Status. Die alten Koordinaten werden wiederhergestellt:

```
10 proc status
20 x=pos(0): y=lin(0)
30 char,0,0,ds$
40 char,x,y
50 endproc status
```

Das nächste Beispiel ist eine abgewandelte Form der Fakultätsdefinition: »fac« ruft sich solange selbst auf (rekursiv), bis die lokale Variable »an« kleiner als 1 geworden ist. Das Endergebnis »er« wird nun an die vorhergehende Prozedur zurückgegeben – bis das Hauptprogamm wieder erreicht ist: »x« in Zeile 70 erhält den Wert des letzten »er«, »n« wurde nicht verändert:

```
10 proc fac(an,& er)
20 if an>0 then er=er*an:'fac(an-1,er)'
30 endproc fac
40 input"Bis wohin multiplizieren";n
50 x=1
```

```
60 'fac(n,x)'
70 print"ergebnis:";x
```

Neue Funktionen von Masterbasic 128

Statt Dezimalzahlen lassen sich jetzt auch hexadezimale und binäre Werte direkt eingeben. Duale Zahlen kennzeichnet man mit <%>, Hexzahlen per <$>.
- Duale Zahlen dürfen maximal 16 Zeichen umfassen (16-Bit-Wert), aber nur 0 oder 1 enthalten.
- Hexzahlen bestehen maximal aus vier Byte (0 bis 9 und A bis F). Sollte die Buchstabenfolge zufällig ein Basic-Befehlswort enthalten, muß man es durch ein Leerzeichen unterbrechen (also statt $DEF3: $DEF3).

v$ = BIN$(x)
... verwandelt die Zahl x (als Zeichenkette) in den entsprechenden Binär-String.
Die Zahlx darf zwischen 0 und 65535 liegen. Ist x kleiner als 256, erhalten Sie nur einen acht Zeichen langen Ergebnis-String (sonst sind's 16).

```
print bin$(15),bin$(256)
000011110000000100000000
```

v = BIN(dual$)
... macht aus dem String »dual$« (ein Binärwert) eine Dezimalzahl zwischen 0 und 65535.
Der Zeichenkette »dual$« darf maximal 16 Zeichen lang sein und nur 0 und 1 enthalten.

```
print bin("10"), bin("10000000")
2 128
```

Vv = BOG(grad)
... rechnet die Gradzahl »grad« ins Bogenmaß um.

```
print bog(180)
3.14159265
```

v = CVAL(ausdruck$)
... wertet die Zeichenkette »ausdruck$« aus.
Der String kann Variablen, Basic-Funktionen oder Berechnungen aller Art enthalten. CVAL ist eine zwar langsamere, aber flexiblere Alternative zu XDEF.

```
10 deffnf(x)=cval(f$)
20 f$="2x":print fnf(4)
30 f$="sin(bog(x))":print fnf(90)
40 a=4: b=20: print cval("a*b")
```

v = DEEK(adresse)
... wirkt wie ein doppeltes PEEK (Lesen von Speicherinhalten). Ersatz für:

```
v=peek(adresse)+peek(adresse+1)*256
```

Dieser Befehl eignet sich ideal, um Programmstart und -ende zu ermitteln:

```
bank15: printdeek($1210)
```

v = DFRE(ga)
... gibt an, wie viele Blöcke noch auf der Diskette in der Floppystation mit Gerätenummer »ga« frei sind. Falls ein DOS-Fehler auftritt, erhält man den Wert - 1.

v = DIV(dividend, divisor)
... ermittelt den ganzzahligen Teil der Division »dividend/divisor«.
Der Wert des Parameters »divisor« darf nur zwischen 1 und 65535 liegen, lediglich »dividend« kann auch 0 sein (sonst gibt's die Fehlermeldung DIVISION BY ZERO).

```
print div(10,3)
3
```

v$ = DUP$(str$,anzahl)
... gibt »anzahl«-mal den String »str$« aus.
Der Maximalwert von »anzahl« darf 255 nicht überschreiten!

```
printdup$("-",40)
printdup$("--",20)
```

... erzeugt eine Linie mit 40 Zeichen Länge.

v = FRAC(x)
... erforscht den Nachkommawert der Realzahl x.

```
print 10/3,frac(10/3)
3.33333333333 0.33333333333
```

v = GRAD(bog)
... verwandelt die Bogenmaß-Zahl in eine Gradzahl.

```
print grad(3.14159265/2)
90
```

v = HIGH(x)
... gibt das High-Byte von x zurück (HIGH(X)= INT(X/256) mit 0 = < X = < 65535).

```
print high(49155)
192
```

v = JA(str$)
Ausgabe der Zeichenkette »str$« an aktueller Cursor-Position inkl. Abfrage (j/n), die man nur mit <J> oder <N> bestätigen kann. Die Funktion erzeugt bei <J> den Wert - 1, sonst 0.
Der Cursor steht anschließend hinter der letzten Klammer.

```
if ja("sicher") then print"wird ausgeführt"
```

v = LG(x[,basis])
... berechnet den Logarithmus von x zur Basis »basis« (voreingestellt ist 10).

```
print lg(10)
1
```

v = LIN(0)
... gibt die aktuelle Zeilennummer des Bildschirms aus.
Der Parameter 0 ist bedeutungslos (Dummy), muß aber angegeben werden.

v = LOW(x)
... bringt das Low-Byte von x (LOW(X)= X AND 255, mit 0 = < X = < 65535.

```
print low(49155)
3
```

v = MAX(arr(0..0))
v = MAX(arr%(0..0))
... ermittelt die höchste Zahl, die im numerischen Array vorhanden ist.

v = MIN(arr(0..0))
v = MIN(arr%(0..0))
... bringt die niedrigste Zahl im numerischen Feld.

v = MOD(dividend,divisor)
... berechnet den Rest der Division aus »dividend/divisor«.
Der Parameter »divisor« darf nur zwischen 1 und 65535 liegen, »dividend« kann auch 0 sein.

```
print mod(10,3)
1
```

v = NEIN(str$)
... ist das Gegenteil der JA-Funktion, d.h. bei <J> wird 0, sonst - 1 übergeben.

v = RANDOM(unter-, obergrenze)
... erzeugt eine Zufallszahl zwischen »untergrenze« und einschließlich »obergrenze«.
Die Unter- und Obergrenzen dürfen zwischen 0 und 65535 liegen, wobei die Untergrenze stets kleiner als die obere sein muß.

```
printchr$(random(65,90))
```

gibt zufällige Buchstaben zwischen A (Code 65) und Z (Code 90) aus.

v = ROUND(x[,nachkommastellen])
... rundet die Zahl x an der vorgesehenen Nachkommastelle (Voreinstellung: 0).

```
print round(3.5), round(-3.1425,3)
4 -3.142
```

v = START(datei[,ga])
Ausgabe der Startadresse des Programms »datei«.
Tritt ein DOS-Fehler auf, erhält die Variable den Wert - 1. Für Geräteadresse »ga« ist 8 voreingestellt.

v = VPEEK(adresse)
... liest den Inhalt von »adresse« im VDC-RAM.

v = VRPEEK(reg)
... holt den Wert des VDC-Registers reg.

```
for i=0 to 31: print vrpeek(i): next
```

... liest alle Register von 0 bis 31.

v = WORD(low,high)
... bildet aus »low« und »high« eine 16-Bit-Zahl.
Die Variablen x und y dürfen nur zwischen 0 und 255 liegen.

```
print WORD(3,192)
49155
```

Datenkonvertierung – vom C 128 zum PC und zurück

Das Tor zur DOS-Welt



Das Laufwerk des C 128 ist in der Lage, PC-Formate zu lesen und zu schreiben. Mit unserem Konvertierungsprogramm DOS Copy können Sie im Büro Ihre Texte wie gewohnt auf einem PC bearbeiten, sie mit nach Hause nehmen und dort auf Ihrem C 128 die Arbeit fortsetzen.

Sie haben in den C64 oder C128 etliche Texte in mühevoller Kleinarbeit eingetippt. Nun sind Sie auf einen PC umgestiegen und haben selbstverständlich nicht die geringste Lust, alles neu einzugeben. Doch leider kann Ihr neues System mit den Disketten vom C64 nichts anfangen: Es benutzt ein völlig anderes Datenformat. Doch es gibt Wege, die alten Texte doch auf den neuen Computer zu überspielen.

Zu Anfang muß jedoch eines gesagt werden: Es lassen sich nur Daten (also Zahlen, Texte etc.) auf einen anderen Computer-Typ übertragen, jedoch keine Programme! Vizawrite z.B. funktioniert nicht auf dem PC.

Das liegt an der vollständig unterschiedlichen Hardware und den Betriebssystemen der einzelnen Computer. Obwohl es für jeden Computer die Programmiersprache Basic gibt, laufen auch diese Programme nicht, da die Syntax des Basic-Interpreters auf dem neuen Computer anders lautet. Sie müssen also auch Basic-Programme um- oder besser völlig neu schreiben.

Drei Wege führen zum Ziel

Doch meistens will man ja gar keine C-64-Programme auf dem PC laufen lassen, sondern nur seine einmal mühsam eingetippten Texte weiterverarbeiten. Um nun die Daten auf den PC zu schaufeln, gibt es drei unterschiedliche Wege:

1. Datenfernübertragung: Die Dateien werden per Telefon und Modem mit Hilfe eines Terminalprogramms ausgetauscht.

2. Direktanschluß des zweiten Computers: Über ein Null-Modem-Kabel lassen sich die Daten direkt über die RS-232-Schnittstelle von einem Computer mit einem speziellen Konvertierungsprogramm zum anderen übertragen.

3. Disketten: Die 1571 kann sowohl PC- als auch C-64-Formate lesen und schreiben. Mit einem geeigneten Hilfsprogramm lassen sich so Dateien relativ leicht konvertieren.

Leider kann man Texte nicht ganz so einfach von einem System auf ein anderes transferieren.

Zeichenkonvertierung

Der Grund liegt in der unterschiedlichen Interpretation der einzelnen Zeichen. Jeder Computer weist jedem Zeichen einen bestimmten Wert, also eine Zahl, zu.

Der C64 und auch der C128 arbeitet mit dem CBM-ASCII (American Standard Code of Information Interchange) (Abb. 2), während der PC den Standard-ASCII Zeichensatz (Abb. 6) benutzt. Da diese Zeichensätze voneinander abweichen, jeder Computer also eine andere Vorstellung besitzt, welchem Zeichen er welchen Wert zuordnet, ist das Chaos schon vorprogrammiert.

Weiterhin codiert jedes Textprogramm die Zeichen, besonders die beim C64 nicht vorhandenen Umlaute, auch noch einmal anders. Der Ziel-Computer ordnet nun den Zeichen eine völlig andere Bedeutung zu. Laut seiner internen Tabelle handelt er zwar richtig, das Ergebnis aber ist ein vollkommen anderes. Deshalb muß in den Übertragungsweg ein Zeichenfilter eingeschleift werden. Dies ist ein kleines Übersetzungsprogramm, das die unterschiedlichen Zeichensätze dem anderen System anpaßt. Hierdurch wird gewährleistet, daß hinterher auch die Buchstaben wieder an der richtigen Stelle auf dem anderen System vorgefunden werden.

Doch die Buchstaben stellen nicht die größte Schwierigkeit dar. Jede Textverarbeitung setzt Steuerzeichen, um z.B. das Zeilenende zu markieren. Da das andere Textverarbeitungsprogramm mit Sicherheit andere Zeichen verwendet, wird nun ein Zeilenende als Buchstabe oder Sonderzeichen erkannt. Der nun auf dem Zielrechner erscheinende Text kann sehr merkwürdig aussehen (Abb. 1). Diesen nun von Hand zu ändern, ist eine äußerst mühevolle Arbeit. Leichter ist es, den Text neu einzugeben.

Welcher Weg zur Datenübertragung auch gewählt wird, jedesmal muß ein Übersetzungsprogramm eingeschaltet werden, das die richtige Zuordnung vornimmt. Dieses Programm sucht sich die entsprechenden Werte aus einer Tabelle, wandelt das übernommene Zeichen und schickt es übersetzt weiter. Hierbei ist es völlig gleichgültig, wo dieses Programm arbeitet. Die Umwandlung kann vor oder nach der Übertragung vorgenommen werden. Nur so ist gewährleistet, daß der Zielrechner hinterher auch wieder den richtigen Text im Speicher stehen hat.

Transfer per Telefon

Bei dieser Art der Datenübertragung ist der Hardwareaufwand naturgemäß größer. Für jeden Rechner wird ein eigenes Modem und ein speziell auf diesen Computer zugeschnittenes Terminalprogramm benötigt. Ein Modem wandelt die vom Computer gelieferten Bits in Töne um, die wie normale Sprache via Telefon übertragen werden können.

An beide Computer werden die Modems angeschlossen, mit der Telefonleitung verbunden, und die Terminalprogramme gestartet. Nach Anwählen der anderen Nummer und Rückmeldung durch den Empfangscomputer kann der Transfer losgehen. Die Verbindung der Rechner über eine Telefonleitung besitzt den Vorteil, daß beide Computer auch an verschiedenen Orten stehen können. Die typische Übertragungsart des Modems begrenzt die Geschwindigkeit, mit der sich Daten übertragen lassen. Damit der C128 alles richtig erkennt, sollten Sie die Transfer-Rate nicht über 300 Baud setzen. Dadurch dauert es natürlich relativ lange, eine größere Datei zum zweiten Computer zu übertragen. Kommt auch noch eine größere Entfernung zwischen den Rechnern hinzu, kassiert die Post kräftig mit. Sie können jedoch auch beide Modems über ein eigenes Kabel miteinander verbinden. Der Datenaustausch wird dadurch zwar nicht schneller, aber wesentlich billiger. Der Betrieb mit der Terminalsoftware funktioniert in beiden Richtungen, so daß Sie genauso Texte vom PC

```
Text View: A:\imine_1                    Col 0              2,427 Bytes          0%
AKTIONEN VON  MINEN__1)  SOCYANATE__ US  MINEN BILDEN SICH MIT  HOSGEN
 SOCYANATE____ SOCYANATE ADDIEREN LEICHT  LKOHOLE, DADURCH WERDEN DIE RECHT
STABILEN
GEBILDET:__ - = =        +         ---    - -  - _ SOCYANAT      LKOHOL        RE
THAN__ LS  RETHANE BEZEICHNET MAN DIE  STER DER  ARBAMINS URE, DIE MAN ALS
 ONDAMID DER  OHLENS URE AUFFASSEN KANN._ IESE GLATTE  DDITIONSREAKTION WIRD
ALS  ERKN PFUNGSREAKTION ZUR  ILDUNG VON  OLYURETHANEN, EINER WICHTIGEN  LASSE
VON  UNSTSTOFFEN ANGEWANDT.  USGEHEND VON  IANINEN WERDEN  IISOCYANATE
GEBILDET.  URCH  EAKTION MIT ZWEIWERTIGEN  LKOHOLEN, BILDEN SICH LINEARE
 OLYURETHAN - OLEK LE.  ERWENDET MAN ABER DREIWERTIGE  LKOHOLE, SO BILDEN SICH
VERNETZTE  OLYURETHANE.  NTERESSANT ABER IST, DA  MAN  OLYURETHAN - CHAUMSTOFFE
DADURCH HERSTELLT, INDEM MAN DER  EAKTIONS MISCHUNG ETWAS  ASSER ZUSETZT.  ACH
DER  EAKTION__ - = =      + - -  ---  -     +    __WIRD      FREIGESETZT,
WODURCH SICH DAS  RODUKT SCHWAMMARTIG AUFBL HT ( OLTOPREN)__2.)  EAKTIONEN VON
 MINEN MIT SALPETRIGER   URE,     __ ALPETRIGE   URE IST SEHR INSTABIL, DESHALB
STELLT MAN SIE IM  EAKTIONSGEMISCH AUS DEM STABILEN  ATRIUMNITRIT MIT
 INERALS URE HER._ RIM RE  MINE GEBEN MIT SALPETRIGER   URE
 IAZONIUMSALZE____ LIPHATISCHE  MINE SPALTEN SOFORT  TICKSTOFF AB.!_ URCH
VOLUMETRISCHE  ESTIMMUNG DER  ENGE DES  TICKSTOFFS KANN MAN DAMIT - ZB. IN
BIOLOGISCHEM ATERIAL - DIE  ENGE PRIM RER  MINOGRUPPEN BESTIMMEN. ( VAN  LYKE -
 EAKTION ).  B PARATIV IST DIESE  EAKTION WENIG INTERESSANT, DA SICH BEI DER
 EAKTION  EMISCHE VON  LKOHOL,  LKYLHALOGENID UND UNGES TTIGTEN
 OHLENWASSERSTOFFEN BILDEN._ ROMATISCHE  IAZONIUMSALZE LASSEN SICH ZWAR ALS
1Help 2Unwrap 3  4Hex  5  6  7Search 8Viewer 9  10Quit
```

[1] Zeichensalat bei fehlender Konvertierungstabelle

ANZEIGE	CHRS	ANZEIGE	CHRS	ANZEIGE	CHRS	ANZEIGE	CHRS
	0	0	48		96	BLK	144
	1	1	49		97	CRSR	145
	2	2	50		98	RVS OFF	146
	3	3	51		99	CLR HOME	147
	4	4	52		100	INST DEL	148
WHT	5	5	53		101		149
	6	6	54		102		150
	7	7	55		103		151
unwirksam SHIFT C=	8	8	56		104		152
wirksam SHIFT C=	9	9	57		105		153
	10	:	58		106		154
	11	;	59		107		155
	12	<	60		108	PUR	156
RETURN	13	=	61		109	CRSR	157
SWITCH TO LOWER CASE	14	>	62		110	YEL	158
	15	?	63		111	CYN	159
	16	@	64		112	SPACE	160
CRSR	17	A	65		113		161
RVS ON	18	B	66		114		162
CLR HOME	19	C	67		115		163
INST DEL	20	D	68		116		164
	21	E	69		117		165
	22	F	70		118		166
	23	G	71		119		167
	24	H	72		120		168
	25	I	73		121		169
	26	J	74		122		170
	27	K	75		123		171
RED	28	L	76		124		172
CRSR	29	M	77		125		173
GRN	30	N	78		126		174
BLU	31	O	79		127		175
SPACE	32	P	80		128		176
!	33	Q	81	Orange	129		177
"	34	R	82		130		178
#	35	S	83		131		179
$	36	T	84		132		180
%	37	U	85	f1	133		181
&	38	V	86	f3	134		182
'	39	W	87	f5	135		183
(	40	X	88	f7	136		184
)	41	Y	89	f2	137		185
*	42	Z	90	f4	138		186
+	43	[	91	f6	139		187
,	44	£	92	f8	140		188
-	45	]	93	SHIFT RETURN	141		189
.	46	↑	94	SWITCH TO UPPER CASE	142		190
/	47	←	95		143		191

[2] Der CBM-ASCII-Zeichensatz

zum C128 schicken können. Diese Art der Datenübertragung funktioniert nicht nur zwischen C128 und PC, sondern zwischen allen Computern, die den Anschluß eines Modems erlauben, also eine serielle Schnittstelle besitzen. Allerdings nur unter der Voraussetzung, daß es sich um ASCII-Dateien handelt.

Convert 64

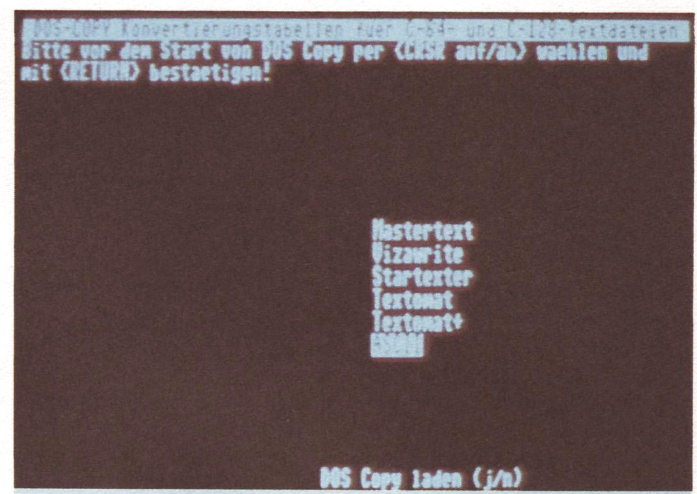
Convert 64 ist ein kombiniertes Transfer- und Zeichensatz-Filterprogramm für den C64 und natürlich auch für den C128 im 64er Modus. Über ein dreiadriges Kabel können Daten zwischen dem C64 und anderen Computern mit normgerechter RS-232-Schnittstelle ausgetauscht werden. Der C64 besitzt am User-Port eine RS-232-Schnittstelle. Die dazu nötige Software ist bereits in seinem Betriebssystem implementiert. Leider liefert der User-Port aber keine normgerechten Pegel für diese Schnittstelle, so daß ihm mit Hilfe eines ICs auf die Sprünge geholfen werden muß. Ohne diesen Zusatz würde bei Anschluß des Verbindungskabels die CIA im C64 bei der ersten Übertragung ihren Geist aufgeben. Mit Hilfe von Convert 64 und der dazugehörigen Hardware können Texte und Dateien zu jedem x-beliebigen anderen Computer übertragen werden. Dabei spielt es keine Rolle, ob es sich um einen PC, Amiga, Atari oder Archimedes handelt. Als wichtigste Vorausetzung muß der angesprochene Computer aber eine Norm-RS-232-Schnittstelle besitzen und ein Terminalprogramm geladen sein. Beide Computer werden über ein dreiadriges Kabel miteinander verbunden. Das Kabel kann eine maximale Länge von 5 m aufweisen. Die Übertragungsrate kann nun auch höher als 300 Baud gewählt werden, da diese Leitung nicht so störanfällig ist, wie eine Telefonleitung.

Das Programm stellt eine komfortable Software zur Verfügung, die auch ein Zeichenfilter für die gängigsten Textverarbeitungsprogramme beinhaltet, wie Vizawrite, Mastertext, Protext, Startexter, sowie CBM-ASCII. Konvertierungstabellen lassen sich aber auch editieren und damit jeder Textverarbeitung auf dem C64 anpassen. Nur Geos-Dateien lassen sich nicht übertragen, da sie nach anderen Kriterien abgelegt sind. Bislang haben wir noch kein Programm gefunden, das diese VLIR-Dateien wandeln kann. Mit diesen Konvertierungstabellen wird sichergestellt, daß am Empfänger-Computer die Texte auch so ankommen, wie sie ursprünglich auf dem C64 zu sehen waren. Das eben Gesagte gilt allerdings nicht für Fettdruck, Kursiv oder sonstige Steueranweisungen für die Druckausgabe.

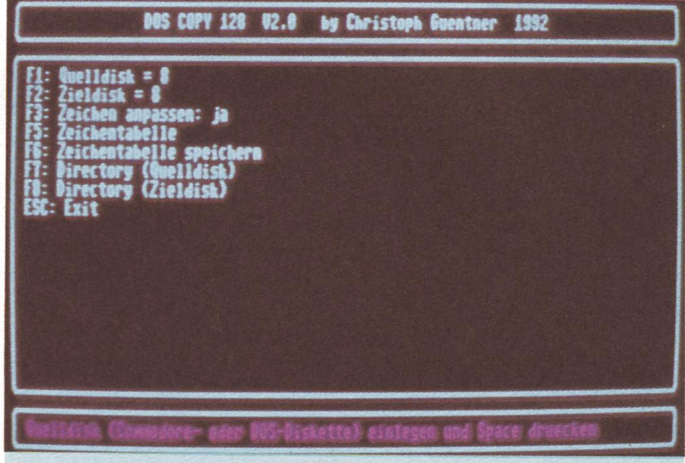
Für den PC, Amiga, Atari oder Archimedes muß nur noch das entsprechende Terminalprogramm vorhanden sein. Achten Sie beim Gebrauch dieser Software darauf, daß bei beiden Computern, also dem C64 und dem Empfangscomputer, die gleichen Übertragungsparameter der RS-232-Schnittstelle eingestellt sind. Ansonsten empfangen Sie Datenmüll, aber keine lesbaren Texte. Keine Angst, bei Experimenten kann die Hardware nicht zerstört werden. Irgendwann haben Sie die optimale Einstellung für Ihr System gefunden und dann flitzen die Texte nur so von einem Computer zum anderen. Convert 64 finden Sie mit Software und ausführlicher Bauanleitung für die RS-232-Schnittstelle im Sonderheft 67.

Daten vom C128 zum PC

Der C128 besitzt ein anderes Laufwerk als der C64. Die 1571 besitzt den gleichen Controller wie ein PC-Laufwerk, deshalb ist sie in der Lage, auch dessen Aufzeichnungsformat zu lesen. Um Daten zwischen den beiden Computern austauschen zu können, brauchen Sie natürlich das schon erwähnte Übersetzungsprogramm. Mit »DOS-COPY« stellen wir Ihnen hier ein einfach zu bedienendes Programm vor.



[3] Diese Textverarbeitungen kann DOS-Copy einwandfrei übersetzen



[4] Das Hauptmenü von DOS-Copy

Das Programm DOS-COPY überträgt beliebige Files von einer Commodore-Diskette auf eine MS-DOS-Diskette und umgekehrt. Voraussetzungen hierfür sind ein C128 mit einer Floppy 1570 oder 1571 und eine formatierte MS-DOS-Diskette.

DOS-COPY ist einfach zu handhaben.

Laden Sie zunächst »DOS COPY.TABINST«. Nach dem Start erscheint ein Menü, in dem Sie per Cursor die gewünschte Konvertierungstabelle (Abb. 3) ins Programm einfügen. Nach »RETURN« und erfolgter Bestätigung werden das eigentliche Hauptprogramm mit den Assembler-Routinen und die Zeichenanpassungstabelle geladen und ein in drei Fenster geteilter Bildschirm erscheint (Abb. 4). Das obere Fenster beinhaltet Name und Versionsnummer des Programms. Das Hauptfenster darunter zeigt links die zur Verfügung stehenden Befehle. Der rechte Bereich ist für das Anzeigen des Directories und Auswählen der zu kopierenden Files. Das untere Fenster dient zum Anzeigen von Meldungen und für Eingaben.

Nun aber genauer zu den einzelnen Menüpunkten. Mit F1 kann die Geräteadresse für das Quellaufwerk, und mit F2 die des Ziellaufwerks geändert werden. Standardeinstellung ist jeweils 8 – ist ein zweites Laufwerk mit der Adresse 9 vorhanden, wird automatisch die Adresse des Ziellaufwerks auf 9 gesetzt. Da die Zeichensätze des C128 und eines PC unterschiedlich sind, müssen (z.B. bei Textfiles) die Zeichen umgewandelt werden, damit der Text dann auf dem PC-Bildschirm richtig erscheint. Der C128 kann viele Grafikzeichen darstellen, die auf einem PC nicht existieren. Dieser kann jedoch z.B. einige griechische Symbole auf den Bildschirm bringen. Die Umwandlung ist jedoch nicht eindeutig umkehrbar, d.h. einigen IBM-Zeichen ist jeweils das gleiche Commodore-

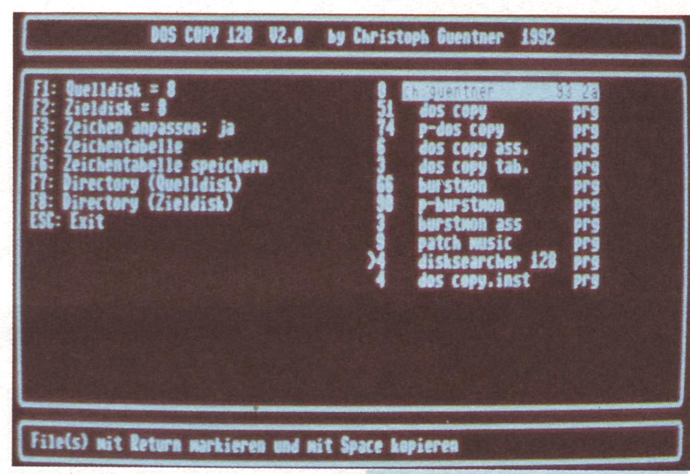
Zeichen zugeordnet. Mit einem Druck auf F3 kann diese Anpassung abgeschaltet werden. Die Umwandlung der Zeichen kann aber auch nach eigenen Wünschen abgeändert werden. Mit F6 erscheint in dem unteren Fenster eine Aufforderung zur Eingabe des ASCII-Codes des Commodore-Zeichens. Nach Übernahme des Wertes mit Return kann nun der ASCII-Code des passenden IBM-Zeichens eingetippt werden. Dann kann das nächste Zeichen eingegeben oder mit ESC die Eingabe beendet werden. Die neu gestaltete Anpassung wird mit F6 gespeichert. Das Directory der Quell- bzw. Zieldisk wird mit F7 bzw. F8 auf dem Bildschirm angezeigt.

Nun aber zur eigentlichen Hauptaufgabe des Programms. Ein Druck auf die Space-Taste und das Directory der eingelegten Diskette wird gezeigt (Abb. 5). Es spielt dabei keine Rolle, ob es sich dabei um eine Commodore- oder eine DOS-Disk handelt. Das Programm analysiert die Diskette und liest dann je nach Format von den jeweiligen Sektoren das Inhaltsverzeichnis. Mit den Cursor-Tasten kann nun im Directory zum gewünschten Eintrag gescrollt werden. Mit Return wird dann dieser markiert. Ist ein File aus Versehen markiert worden, so kann dies mit Return rückgängig gemacht werden. Shift + Home entfernt alle Markierungen. Bei einer MS-DOS-Disk werden normale Files mit Kleinbuchstaben und Subdirectories in Großbuchstaben angezeigt. Wird nun ein Subdirectory mit Return markiert, so wird dieses eingelesen und angezeigt. Alle vorigen Markierungen werden dabei gelöscht. Sind nun alle Files, die kopiert werden sollen, ausgewählt, beginnt der Kopiervorgang mit einem erneuten Betätigen der Space-Taste. Nachdem alle Files eingelesen sind oder für das nächste File nicht mehr genügend Speicher frei ist, erscheint eine Aufforderung (bei Verwendung zweier Laufwerke entfällt diese), daß die Zieldiskette einzulegen ist. Ist dies geschehen, so bewirkt wiederum das Drücken der Space-Taste, daß die Files auf die Zieldiskette geschrieben werden – (die Zieldiskette muß formatiert sein!). Auch hier stellt das Programm selbständig fest, um welche Art von Diskette es sich handelt. Ist nicht genügend Platz auf der Diskette, wird eine Meldung ausgegeben und der Kopiervorgang beendet. Existiert das File bereits auf der Diskette, besteht die Möglichkeit, den Namen des zu speichernden Files zu ändern oder mit Return dieses nicht zu speichern. Wurden alle Files kopiert, ist ein erneutes Sichern der Files auf eine weitere Diskette mit »s« möglich.

Es ist also Kopieren von Commodore-Disk auf Commodore-Disk, Commodore-Disk auf DOS-Disk sowie DOS-Disk auf DOS-Disk möglich. Die DOS-Disk darf dabei nicht mehr als 40 Spuren pro Seite aufweisen, damit sie gelesen werden kann. Beim Arbeiten mit einer 1570 dürfen außerdem nur einseitige Disketten benutzt werden. Es werden also Disketten mit bis zu 360 KByte (bzw. 180 KByte bei der 1570) akzeptiert. Dies entspricht 40 Spuren mit jeweils neun Sektoren 512 Byte. Besitzt man einen PC mit einem High-Density-(HD-) Laufwerk für das 1,2-MByte-Format, so sind die Disketten mit »format a: /4« (bzw. »format a: /4/1« für die 1570) zu formatieren, damit diese im 40-Spur-Format beschrieben werden. Üblich ist das 360-KByte-Format, aber auch das alte 320-KByte-Format werden von DOS-COPY gelesen und geschrieben.

Nun steht einem Datenaustausch nichts mehr im Wege.

(Güntner/jh)



[5] Der gewünschte Text wird mit dem Cursor ausgewählt, mit RETURN bestätigt und die Space-Taste startet den Konvertierungsvorgang

ASCII-Tabelle

Der Standard-ASCII-Zeichensatz

Dez.	Hex	Zeichen	Dez.	Hex	Zeichen	Dez.	Hex	Zeichen	Dez.	Hex	Zeichen	Dez.	Hex	Zeichen	Dez.	Hex	Zeichen	Dez.	Hex	Zeichen	Dez.	Hex	Zeichen
0	00		32	20		64	40	@	96	60	`	128	80	Ç	160	A0	á	192	C0	└	224	E0	α
1	01	☺	33	21	!	65	41	A	97	61	a	129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
2	02	☻	34	22	"	66	42	B	98	62	b	130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
3	03	♥	35	23	#	67	43	C	99	63	c	131	83	â	163	A3	ú	195	C3	├	227	E3	π
4	04	♦	36	24	$	68	44	D	100	64	d	132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
5	05	♣	37	25	%	69	45	E	101	65	e	133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
6	06	♠	38	26	&	70	46	F	102	66	f	134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
7	07	•	39	27	'	71	47	G	103	67	g	135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
8	08	◘	40	28	(	72	48	H	104	68	h	136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
9	09	○	41	29	)	73	49	I	105	69	i	137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
10	0A	◙	42	2A	*	74	4A	J	106	6A	j	138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
11	0B	♂	43	2B	+	75	4B	K	107	6B	k	139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
12	0C	♀	44	2C	,	76	4C	L	108	6C	l	140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
13	0D	♪	45	2D	-	77	4D	M	109	6D	m	141	8D	ì	173	AD	¡	205	CD	═	237	ED	φ
14	0E	♫	46	2E	.	78	4E	N	110	6E	n	142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
15	0F	—	47	2F	/	79	4F	O	111	6F	o	143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
16	10	►	48	30	0	80	50	P	112	70	p	144	90	É	176	B0		208	D0	╨	240	F0	≡
17	11	◄	49	31	1	81	51	Q	113	71	q	145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
18	12	↕	50	32	2	82	52	R	114	72	r	146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
19	13	‼	51	33	3	83	53	S	115	73	s	147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
20	14	¶	52	34	4	84	54	T	116	74	t	148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
21	15	§	53	35	5	85	55	U	117	75	u	149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
22	16	▬	54	36	6	86	56	V	118	76	v	150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
23	17	↨	55	37	7	87	57	W	119	77	w	151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
24	18	↑	56	38	8	88	58	X	120	78	x	152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	•
25	19	↓	57	39	9	89	59	Y	121	79	y	153	99	Ö	185	B9	╣	217	D9	┘	249	F9	•
26	1A	→	58	3A	:	90	5A	Z	122	7A	z	154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
27	1B	←	59	3B	;	91	5B	[	123	7B	{	155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
28	1C	∟	60	3C	<	92	5C	\	124	7C	¦	156	9C	£	188	BC	╝	220	DC	▄	252	FC	η
29	1D	↔	61	3D	=	93	5D	]	125	7D	}	157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
30	1E	▲	62	3E	>	94	5E	^	126	7E	˜	158	9E	₧	190	BE	╛	222	DE	▐	254	FE	•
31	1F	▼	63	3F	?	95	5F	_	127	7F	⌂	159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

[6] Der ASCII-Zeichensatz

Kurzinfo: DOS-Copy

Programmart: Datenkonvertierung
Bildschirmmodus: 80 Zeichen (VDC)
Laden: DLOAD "DOS COPY.TABINST"
Starten: nach dem Laden RUN eingeben
Besonderheiten: konvertiert C-128-Dateien auf PC-Format und umgekehrt
Benötigte Blöcke: 82
Programmautor: Christoph Güntner

Disk-Monitor

Floppy auf Trab gebracht

Viele Fähigkeiten des C 128 und seiner Floppy liegen brach. Wir erwecken mit dem »Burstmon« einige davon zum Leben. Damit wird die Floppy zum Tausendsassa.

Der C128 ist schneller als der C64, deshalb spendierte man ihm auch ein schnelleres Floppy-Laufwerk, die 1571 (Abb.). Um möglichst kompatibel zum 1541-Standard zu bleiben, besitzt dieses Laufwerk zwei unterschiedliche Betriebsarten, in denen es entweder langsam wie sein Vorfahr arbeitet oder aber im schnellen Modus einen Zahn zulegt. Wenn es an einen C128 angeschlossen ist, wird es beim Einschalten des Computers automatisch in den schnellen Modus umgeschaltet, beim C64 hingegen arbeitet es ausschließlich in der langsameren Betriebsart.

In der Turbo-Betriebsart stehen interessante zusätzliche Befehle zur Verfügung, die alle mit den Zeichen »U0« beginnen. Mit ihrer Hilfe lassen sich nicht nur sämtliche Floppybefehle ausführen, sondern beispielsweise auch Fremdformate bearbeiten. Der Weg zum IBM-Format steht somit offen.

Um mit diesen Befehlen arbeiten zu können, benötigen Sie als Werkzeug den »Burstmon«, einen C128 mit RGB-Monitor im 80-Zeichen-Modus und natürlich die 1571.

Laden Sie das Programm mit

```
RUN "Burstmon"
```

Es erscheint das Hauptmenü, in dem Sie mit den Funktionstasten die einzelnen Befehle anwählen können. Da die U0-Befehle nicht auf das Commodore-Format beschränkt sind, können Sie auch zahlreiche Fremdformate bearbeiten, z. B. PC- und CP/M-Disketten. Arbeiten Sie aber sicherheitshalber immer mit Kopien. Da Sie mit dem Burstmon auch den Disketteninhalt verändern können, besteht die Möglichkeit, daß Sie sich ihre Daten, Programme oder was immer auf der Diskette ist, zerstören.

Bildschirm-Meldungen

Im oberen Teil des Bildschirms zeigt das Programm den Controllerstatus, der aus dem »Burst-Statusbyte« gewonnen und im Klartext angezeigt wird. Außerdem erhalten Sie hier Informationen über

Track	Nummer der aktuellen Spur
Sector	Nummer des aktuellen Sektors
Format	Format der letzten bearbeiteten Diskette
Sector width	Größe der physikalischen Sektoren
Capacity	maximale Sektoranzahl der aktuellen Spur

Bei Fehlern erscheint das Statusfeld in roter Schrift, bei fälligen Eingaben in Gelb, sonst in der Farbe Cyan.

Im unteren Bereich befindet sich das Eingabe-Fenster. Außerdem erfolgt hier die Ausgabe des Floppy-Status (entspricht dem Inhalt des Fehlerkanals), den man nicht mit dem bereits erwähnten Controller-Status verwechseln darf.

Mit dem Klammeraffen können Sie hier die normalen Floppy-Befehle, also beispielsweise zum Formatieren oder Löschen von Dateien, eingeben. Der Klammeraffe allein zeigt den Disk-Status an.

Wenn Sie »EDIT« eingeben, gelangen Sie direkt in den Editor, mit dem Sie einzelne Bytes eines Diskettensektors ändern können. Diesen Modus beenden Sie mit dem <-Zeichen.

Durch Eingabe von »X« können Sie den Burstmon verlassen.

Nun zu den Befehlen, die mit Funktionstasten erreichbar sind:

F1 Read

Damit können Sie einzelne Sektoren von der Diskette lesen. Nachdem Sie diese Funktion gewählt haben, müssen Sie Spur- und Sektornummer eingeben (dezimal). Nachdem die Floppy den kompletten Sektor gelesen hat, wird er auf dem Bildschirm angezeigt: Links finden Sie die Byte-Nummer, im mittleren Bildschirmbereich den Sektorinhalt im Hex-Format und rechts daneben als ASCII-Zeichen. Den blinkenden Cursor können Sie mit den Cursor-Tasten bewegen und Änderungen im Hex- und ASCII-Teil vornehmen. Unerlaubte Zeichen werden hierbei ignoriert.

Der Befehl arbeitet nur dann korrekt, wenn Sie die Diskette vorher einmal mit dem »Inquire«-Befehl angemeldet haben. Dies ist notwendig, damit sich Programm und Floppy auf das jeweilige Diskettenformat einstellen können.

Um den nächsten bzw. vorherigen Sektor anzeigen zu lassen, drücken Sie einfach <+> bzw. <->.

Mit der ESC-Taste können Sie den Monitor-Modus wieder verlassen und ins Hauptmenü gelangen.

Burst-Befehle

Burstbefehle beginnen immer mit der Byte-Sequenz $55, $30 (entspricht den Zeichen U0).

Dann folgt mindestens ein Byte mit diesem Aufbau:

Bit0: Drivenummer (immer 0)

Bit1 bis 3: Befehlscode, gibt die Art des Befehls an
- 010: Inquire
- 110: Formatieren MFM
- 110: Formatieren GCR
- 001: Interleave
- 101: Query

Bit 4: Seitennummer

Bit 5 und 6: ohne Bedeutung

Bit 7: Flag für besondere Angaben (Formatierung, Interleave neu setzen u.ä.)

Anschließend können noch Parameter für die Formatangaben (Sektorgröße, -versatz, Spurnummern etc.) folgen.

F2 Write

Mit diesem Befehl können Sie einen im Speicher vorhandenen Sektor wieder auf die Diskette schreiben. Achtung: Wenn Sie Änderungen vorgenommen haben, können Sie sich damit einzelne oder alle Daten zerstören!

F3 Directory

Dies zeigt das Inhaltsverzeichnis einer Diskette an.

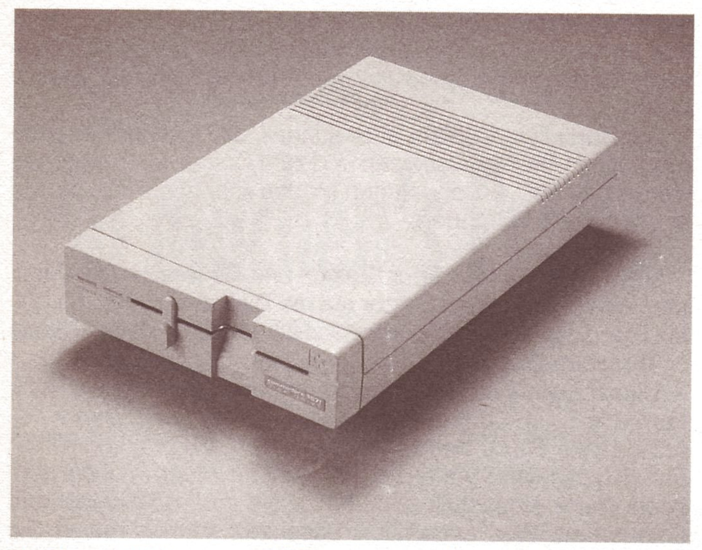
F4 Status

zeigt den Disk-Status im Eingabe-Fenster an. Wie bei allen Befehlen, die eine neue Statusmeldung im oberen oder unteren Bildschirmteil anzeigt, wartet Burstmon auf den nächsten Tastendruck, bis er seine Arbeit fortsetzt.

Den Status können Sie sich auch durch Eingabe des Klammeraffen @ anzeigen lassen.

F5 Inquire

Mit diesem Kommando initialisieren Sie eine neueingelegte Diskette. Dieser Befehl muß nach jedem Diskettenwechsel einmal ausgeführt werden.



Die Floppy 1571 kann am C128 auch sehr schnell arbeiten

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte, wobei die Bits folgende Bedeutung haben: Bit 0: Drivenummer (immer 0) Bit 1: immer 1 Bit 2: immer 1 Bit 3: immer 0 Bit 4: Seitenummer der Diskette 0 oder 1 Bit 5: Flag für Formatieren; 1=beidseitig; 0=einseitig Bit 6: Flag für Index Address Mark; 1= schreiben; 0= nicht schreiben Bit 7: immer 0 bei MFM
03	Steuerbyte für Formatierung: Bits 0 bis 5 enthalten den logischen Startsektor Bit 6: Anzeige der Sektortabelle: 1=Sektortabelle wird vom Computer übergeben 0= Sektortabelle wird von der Floppy selbst erstellt. Bit 7: Flag für Formatierung; bei MFM-Format immer 1
04	Sektorabstand für Hardware (sector interleave) (Angabe nicht notwendig); bei fehlender Angabe 0
05	Größe der Sektoren (0-3): (Angabe nicht notwendig) 0—128 Bytes pro Sektor 1—256 Bytes pro Sektor (bei fehlender Angabe) 2—512 Bytes pro Sektor 3—1024 Bytes pro Sektor
06	Letzte Spurnummer beim Formatieren; Angabe nicht notwendig; sie wird dann 39 gesetzt
07	Anzahl der Sektoren pro Spur (Angabe nicht notwendig). Wird der Parameter angegeben, so richtet sich die Maximalzahl der Sektoren nach Byte 05 und zwar: 26 bei Größe 0 16 bei Größe 1 9 bei Größe 2 5 bei Größe 3
08	Nummer der Spur für Start der Formatierung (Angabe nicht notwendig; Floppy setzt dann 0)
09	Physikalischer Beginn der ersten Spur: (Angabe nicht notwendig; Floppy setzt dann 0)
10	Füllbyte; Wert, mit dem die Sektoren beim Formatieren gefüllt werden sollen; (Angabe nicht notwendig; die Floppy setzt dann $e5 als Defaultwert)
11-...	Bytes der Sektortabelle, falls in Byte 03 des Befehls das Bit 6 gesetzt war.

Busverkehr: Nach dem Senden des Befehls und der Daten vom Computer über den Kommandokanal kein Busverkehr mehr. Die Floppy sendet beim Formatieren kein Statusbyte!

Der Aufbau des Format-Befehls: Hiermit sind die unterschiedlichsten Diskettenformate möglich

F6 Query

Dieses Kommando kann als Ersatz für Inquire verwendet werden, dauert jedoch etwas länger, da die Diskette auf ihr Format hin untersucht wird. Somit ist Query bei Disketten unbekannter Herkunft sehr nützlich.

Sie müssen dem Programm mitteilen, welche Spur geprüft werden soll, es ist theoretisch durchaus möglich (aber glücklicherweise sehr selten), daß die Spuren einer Diskette unterschiedliche Formate aufweisen. Anschließend zeigt Burstmon die Spurnummer, die Anzahl der darin enthaltenen Sektoren, deren kleinste und größte Nummer (die Numerierung muß nicht bei 0 oder 1 beginnen) sowie den Interleave-Faktor (Textkasten) an.

F7 BAM-Analyse

Wenn Sie diesen Punkt benutzen, haben Sie vier Wahlmöglichkeiten:

- F1: BAM-Belegung anzeigen

Die in der BAM (Block Alocation Map, Blockbelegungstabelle) als benutzt gekennzeichneten Sektoren werden grafisch auf dem Bildschirm angezeigt

- F3: Sektoren einer Datei anzeigen

Nach der Eingabe des Dateinamens werden alle zur Datei gehörenden Sektoren in der logischen Reihenfolge spurweise angezeigt. Nachdem Sie eine weitere Taste gedrückt haben, wird ein BAM-Plan angezeigt, in dem nur die von der Datei benutzten Blöcke farblich hervorgehoben sind.

- F5: analysierte Files anzeigen

gibt die Namen der bereits bearbeiteten Dateien aus.

- F7: Directory

zeigt Ihnen das Inhaltsverzeichnis einer Commodore-Diskette an.

Mit <X> verlassen Sie die BAM-Analyse.

Nun kommen wir zu den mächtigsten Befehlen, den U0-Anweisungen.

Interleave – was ist das?

Unter einem Interleave-Faktor versteht man den Abstand, der zwischen zwei logisch aufeinanderfolgenden Sektoren einer Spur liegt. Da eine Diskette recht schnell rotiert, folgen die Bytes auch rasch aufeinander. Wenn nun auf den Sektor 1 unmittelbar der Sektor zwei folgen würde, hätte die Elektronik der Floppy nicht genügend Zeit, den kompletten Sektor zu bearbeiten und weiterzutransportieren.

Aus diesem Grund haben die Sektoren einen gewissen Versatz, den Interleave, d.h., sie folgen nicht in der Reihenfolge 1-2-3-4... sondern bei Interleave 3 1-4-7-10...

Dadurch gewinnt die Floppy Zeit, die Zeit, in der die nicht benötigten Sektoren am Kopf vorbeiziehen. Im Idealfall ist der benötigte Sektor genau dann wieder vor dem Lesekopf, wenn die Floppy ihre Arbeit gerade beendet hat.

F8 User0

Hier steht Ihnen wieder ein Menü mit zehn Punkten zur Verfügung, mit dem Sie die schnellen sog. Burst-Befehle verwenden können. Da dies nur in Maschinensprache mit dieser Geschwindigkeit möglich ist, sollten Sie sich, falls Sie sie in eigenen Programmen verwenden möchten, mit Maschinensprache auskennen. Der prinzipielle Aufbau eines solchen Kommandos ist einfach: Es besteht aus mindestens 3 Byte. Die beiden ersten enthalten die Werte $55 (85) und $30 (48). Dies entspricht den ASCII-Codes U0. Das dritte Byte kennzeichnet den eigentlichen Befehl (Aufbau siehe Textkasten).

Im Burstmon brauchen Sie sich um diese Probleme nicht zu kümmern, hier nimmt Ihnen das Programm die Arbeit ab. Sie können aus folgenden zehn Befehlen wählen:

1) Burst Read

Wie im Hauptmenü lesen Sie hiermit den gewünschten Sektor von der Diskette. Dazu müssen Sie Spur- und Sektornummer und die Diskettenseite (0 oder 1) eingeben. Anschlie-

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte, wobei die Bits folgende Bedeutung haben:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 1
	Bit 2: immer 1
	Bit 3: immer 0
	Bit 4: egal
	Bit 5: egal
	Bit 6: egal
	Bit 7: Flag für Formatierung: Hier kann gewählt werden ob eine Spur vor dem Formatieren ausgemessen wird oder nicht. Das Ausmessen sorgt für eine gleichmäßige Verteilung aller Sektoren und entsprechend gleich großen Lücken zwischen den einzelnen Sektoren. 1 bedeutet Spur ausmessen, 0 bedeutet Spur nicht ausmessen
03	Steuerbyte für Formatierung: bei GCR immer 0
04	erstes Zeichen der ID
05	zweites Zeichen der ID
Busverkehr: Auch hier gilt wieder, daß die Floppy kein Burst-Statusbyte ausgibt, nachdem die Diskette formatiert wurde.	

Beim GCR-Format hat man keine Einstellungsmöglichkeiten

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; wobei die Bits folgende Bedeutung haben:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 0
	Bit 2: immer 0
	Bit 3: immer 1
	Bit 4: immer 0
	Bit 5: egal
	Bit 6: egal
	Bit 7: Flag für Schreiben oder Lesen des Interleave: 0= Interleave wird neu gesetzt, 1= Interleave wird von der Floppy ausgegeben
03	Interleave-Wert (nur, wenn 02/Bit 7 gleich 0)
Busverkehr: Wird Interleave neu gesetzt, so muß der Computer das Byte hinter den Kommandobytes an die Floppy senden. Die Floppy gibt daraufhin kein Byte aus. Wird der Interleave-Wert vom Computer gelesen, so sendet die Floppy nach Erhalt des Kommandos ein Byte mit dem Interleave-Wert.	

Der Interleave-Faktor bestimmt den Abstand zweier Sektoren und bestimmt die Geschwindigkeit

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; die Bits haben folgende Bedeutung:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 1
	Bit 2: immer 0
	Bit 3: immer 1
	Bit 4: Diskettenseite im MFM-Format
	Bit 5: egal
	Bit 6: egal
	Bit 7: Flag für folgende Spurnummer: bei 1 folgt noch Byte 03 mit Spurnummer
03	Spurnummer (angeben, wenn 02/Bit 7 gleich 1 ist)
Busverkehr: Nach Beenden des Kommandos gibt die Floppy ein Burst Statusbyte aus. Wurde ein MFM-Format festgestellt, so folgen zusätzlich Informationen.	

Der Query-Befehl meldet das Diskettenformat bei der Floppyelektronik an

ßend befindet sich Burstmon wieder im Editor-Modus, in dem Sie Änderungen vornehmen können. Das Vor- und Rückwärtsblättern mit <+> und <-> funktioniert hier jedoch nicht.

Der Burst-Read-Befehl ist so schnell, daß in einer Sekunde mehr als 21 Sektoren gelesen und zum Computer übertragen werden können. Im langsamen Modus schafft die Floppy das in etwa zwölf Sekunden.

2) Burst Write

Nach Angabe von Spur, Sektor und Seite schreibt Burstmon den bearbeiteten Sektor auf die Diskette zurück. Es gelten hier die gleichen Warnungen wie im Normal-Write-Modus.

3) Inquire Disk

Dieser Befehl dient wieder zum Initialisieren der Diskette, damit der Disk-Controller über das Format der Diskette Bescheid weiß. Da es, besonders bei den MFM-Formaten (CP/M und PC) sehr viele Varianten gibt, muß die Floppy auf die richtigen Werte eingestellt sein, bevor sie einen Schreibzugriff unternimmt. Was andernfalls passieren könnte, malen Sie sich besser selbst aus.

4) Format MFM

Dies ist ein sehr mächtiger Befehl, Sie können praktisch alle Diskettenformate herstellen. Allerdings wird hier nur physikalisch formatiert. Directory- und BAM-Einträge werden nicht angelegt. Burstmon fragt Sie nach den einzelnen Format-Parametern.

5) Format GCR

Hiermit können Sie eine Diskette im Commodore-Format formatieren. Übrigens ist eine neue formatierte Diskette nicht automatisch initialisiert. Sie müssen also vor der weiteren Bearbeitung noch den Inquire- oder Query-Befehl aufrufen.

6) Sector Interleave

Hiermit können Sie den Abstand zwischen zwei logisch aufeinanderfolgenden Blöcken einer Spur lesen oder eine Voreinstellung vornehmen. Zur näheren Bedeutung des Interleave s. Textkasten.

7) Query Disk Format

Wie im Hauptmenü wird hier die Diskette initialisiert und auf ihr Format untersucht. Die Parameter werden anschließend angezeigt.

8) Inquire Status

Mit diesem Befehl können Sie das Burst-Status-Byte, das die Floppy bereitstellt, anzeigen lassen.

9) Utility Mode

Hier stehen Ihnen wieder mehrere Befehle zur Wahl zur Verfügung, mit denen Sie
- den Interleave für GCR-Disketten einstellen
- die Anzahl an Leseversuchen festlegen
- die ROM-Kennung überprüfen
- zwischen 1541- und 1571-Modus umschalten
- die Diskettenseite einstellen und
- die Geräteadresse umstellen.

Mit <X> verlassen Sie dieses Untermenü.

Die Umstellung der Geräteadresse betrifft nur die Floppy. Wenn Sie diese also auf 9 setzen, greift Burstmon immer noch auf Gerät 8 zu.

Um Burstmon auf andere Adressen zugreifen zu lassen (erlaubt sind 8 bis 12), geben Sie im Hauptmenü den Klammeraffen und danach die neue Adresse ein. (hb)

Kurzinfo: Burstmon

Programmart: Diskettenmonitor
Bildschirmmodus: 80 Zeichen (VDC)
Laden: RUN "Burstmon"
Starten: automatisch
Besonderheiten: arbeitet mit allen MFM-Formaten
Benötigte Blocks: 69
Programmautor: Karsten Schramm / Christoph Güntner

Tips und Tricks

zu Mastertext 128 V2.0

So machen's Profis

Leicht zu bedienen: Mastertext V2.0, das beliebteste Textverarbeitungsprogramm für den C 128. Wir zeigen Ihnen, wie man die ideale Arbeitsdiskette zusammenstellt – damit man die Original-Systemdisk im Panzerschrank lassen kann!

Mit Mastertext lassen sich Textdokumente aller Art erzeugen, z.B. Rechnungen, Angebote, Manuskripte usw. Zu jeder dieser Textgruppen kann man Formblätter, Tabulatoren mit Floskeltexten, Druckerparameter (für Schriftarten, Geräte- und Sekundäradressen) oder Wörterbücher für Master-Spell zusammenstellen und als Parameter-Dateien speichern.

Aber bei der täglichen Arbeit mit dem Textverarbeitungsprogramm geht's schon los: Mastertext laden, Diskette wechseln, Formblatt, Tabulatoren, Druckerparameter, Wörterbücher und Mustertexte aus der Diskettensammlung herausfischen und laden – alles verbunden mit umständlicher, zeitraubender Suche. Und zu guter Letzt hat man dann doch vergessen, eine wichtige Datei in den Computer zu holen! Ab sofort soll das anders werden: Jetzt wird profi-like gearbeitet!

- Formatieren Sie für jede vorgesehene Textgruppe (z.B. »Rechnungen«) eine separate Diskette (wer die Floppy 1571 benutzt, sollte für die 1328 Blocks am besten DS/DD-Disketten verwenden). Tragen Sie als Header den gewünschten Namen ein. Wer deutsche Umlaute oder Sonderzeichen für den Diskettennamen braucht, sollte Disketten mit der Option »Floppy« formatieren (per ESC-Taste im Text-Editor aufzurufen): Dann erscheinen auch die Umlaute auf dem Bildschirm.
- Kopieren Sie auf die erste Profi-Diskette folgende Dateien (von Ihrer bisherigen, per »Mastertext/Inst« angepaßten System-Disk):
- MASTERTEXT V2.0,
- MASTERTEXT T1,
- MASTERTEXT T2,
- MASTERTEXT T3,
- MASTERTEXT T4,
- MASTERTEXT T5,
- ZEICHENSATZ,
- TYP.
- Sie laden und starten Mastertext und holen von der bisherigen Arbeitsdisk die gespeicherten Kostbarkeiten zur Textgruppe, z.B. Formulare oder Texte und speichern sie mit markantem Dateinamen (FORMULAR mit Suffix »f«) auf die neue Profi-Diskette. Es erscheint die Meldung »file exists« (mit <RETURN> bestätigen!). Die anschließende Frage »ersetzen j/n?« muß man bejahen.
- Machen Sie's mit den Dateien für Tabulatoren (Endung »b«) und Druckerparameter (Suffix »d«) ebenso.
- Wer will, kann die Profi-Diskette noch mit dem individuellen Master-Spell-Wörterbuch (Endung »w«) zu dieser Textgruppe ergänzen.

Auf unserer ersten Mastertext-Profi-Disk sind etwa 189 Blocks belegt (ohne Wörterbuch-Datei): 1139 freie Sektoren oder knapp 290 KByte stehen zum Speichern beliebiger Textdateien zur Verfügung. Vorbei ist's mit lästigen Diskettenwechseln: Alle benötigten Files findet Mastertext auf Ihrer Profi-Disk, einem Zwitter aus System- und Arbeitsdiskette. Enger wird's, wenn man DFÜ betreiben oder Rundschreiben versenden möchte – dann muß man auch die entsprechenden Files der alten Arbeitsdiskette übernehmen.

Boot-Sektor installieren

Von jeder Profi-Disk zu gewählten Textgruppen sollte man Mastertext per Reset oder mit der Anweisung BOOT starten.

Dazu findet man auf der Test-Demo-Disk zur Floppy 1571 (Gratiszugabe beim Kauf) den »Autoboot-Maker«. Wer aber die Programmservice-Diskette zum 128er Sonderheft 29 besitzt, entdeckt auf der Rückseite das Tool »Boot-Sektor«, mit dem's bedeutend eleganter geht. Nach dem Laden (DLOAD), aber vor dem Start mit RUN, sollten Sie die vorgesehene Mastertext-Profi-Disk einlegen – die Fragen erscheinen:

- Text hinter Booting: Tragen Sie z.B. den Namen der Profi-Disk (Vorträge, Rechnungen usw.) ein (oder einfach <RETURN>).
- Name des zu ladenden Maschinen-Programms: ...übergeht man ebenfalls mit <RETURN>.
- Name des zu ladenden Basic-Programms: MASTERTEXT V2.0,
- Soll das Basic-Programm automatisch gestartet werden?: ...beantwortet man mit <J>, ebenso die folgende Sicherheitsabfrage (Richtige Diskette im Laufwerk?). Der Boot-Sektor wird auf Disk implementiert. Machen Sie die Probe aufs Exempel: Lassen Sie die Diskette im Laufwerk und drücken Sie den Reset-Knopf (oder geben Sie BOOT ein) – das Textprogramm wird jetzt geladen und gestartet.

Noch ein Sahnehäubchen drauf? Ideal ist, wenn Mastertext beim Booten bereits im Titelbild einen entsprechenden Hinweis auf den Textgruppen-Typ der eingelegten Profi-Diskette bringt, z.B. »Rechnungen Mai 1993« usw.

Laden Sie die Datei »Mastertext V2.0« von Ihrer Profi-Diskette (per DLOAD, nicht RUN!) und LISTen Sie Zeile 160. Dort ist der Originaltext »Loading...« untergebracht, den man nach Belieben verändern kann. Vorher sollten Sie per <SHIFT CBM> zum Klein- und Großschriftzeichensatz umschalten.

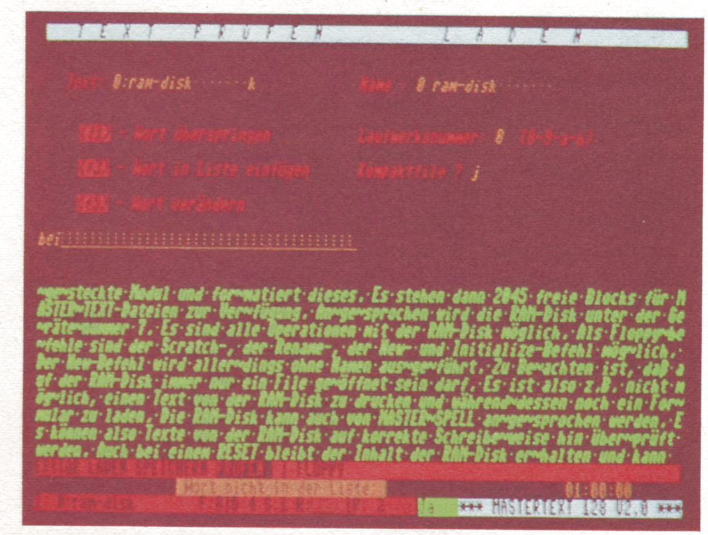
Tilgen Sie die alte Ladedatei von Diskette:

```
SCRATCH "MASTERTEXT V2.0"
```

und speichern Sie die geänderte Version:

```
DSAVE "MASTERTEXT V2.0"
```

Jetzt läßt sich die Profi-Disk mit der neuen Einschaltmeldung booten. (Erich Wüstner/bl)



Textverarbeitung mit professionellen Funktionen: Mastertext 128 V2.0

Tips zu Superscript 128

Schnell und seriell

Viele C-128-User arbeiten mit Superscript, dazu haben wir eine Tastatur-Kurzreferenz zusammengestellt und Tips für seriell angeschlossene Drucker.

Was aber tun, wenn man einen Drucker besitzt, der lediglich mit einem seriellen Kabel ausgestattet ist, z.B. der Star LC-10C oder andere Epson-kompatible, die per Interface am C128 angeschlossen sind? Dann bleibt nicht anderes übrig, als das File »Defaults« von der Superscript-Systemdisk zu laden (Taste <F2> im Hauptmenü) und die entsprechenden Werte einzutragen. Das geht wie beim Verbessern z.B. eines Briefs: Bytes ändern und wieder speichern. Auf der Diskette zu diesem Sonderheft finden Sie eine Defaults-Datei, die speziell für den 24-Nadler Star LC 24 mit Centronics-Schnittstelle und Userport-Kabel eingerichtet wurde. Zum Betrieb mit dem seriell verbundenen Star LC-10C muß man die Daten entsprechend ändern (s. Tabelle) und die Treiberdatei auf die Systemdiskette von Superscript kopieren.

Die erforderliche Einstellung der DIP-Schalter:
- 1, 2, 3, 6, 8 und 10: on,
- 4, 5, 7, 9: off.

(Else Goerke/bl)

Superscript 128 (Tastaturfunktionen)

Kontrollanweisungen	
<CRTL A>	Block anhängen
<CTRL B>	Textanfang
<CTRL C>	Eingabe löschen
<CTRL D>	Wort löschen
<CTRL E>	nächstes Wort
<CTRL F>	zwischen Groß- und Kleinschrift wechseln
<CTRL G>	Textende
<CTRL H>	Cursor links
<CTRL I>	Tabulator
<CTRL J>	Cursor abwärts
<CTRL K>	Cursor aufwärts
<CTRL L>	Gestaltungszeichen
<CTRL M>	wie RETURN-Taste
<CTRL N>	neue Zeile
<CTRL O>	vorhergehender Absatz
<CTRL P>	nächster Absatz
<CTRL Q>	Aktion beenden
<CTRL R>	Befehl wiederholen
<CTRL S>	Leerzeichen einfügen
<CTRL T>	Zeile abschneiden
<CTRL U>	Cursor rechts
<CTRL V>	Einfüge-Modus ein/aus
<CTRL W>	vorhergehendes Wort
<CTRL X>	Dokument neu formatieren
<CTRL Y>	Absatz neu formatieren
<CTRL Z>	vorhergehende Zeile
<CTRL #>	Tabulator rückwärts
Spezialtasten	
<RUN/STOP>	Befehlssequenz wählen
<ESC>	Befehlssequenz wählen
<TAB>	Tabulatorsprung vorwärts
<SHIFT/TAB>	Tabulatorsprung rückwärts
<HELP>	Hilfsbildschirm
<CLR>	Dokumentanfang
<HOME>	Bildschirmanfang
<INST>	ein Zeichen einfügen
<DEL>	ein Zeichen löschen
Funktionstasten	
<F1>	Hauptmenü
<F2>	Dokument laden
<F3>	vorhergehendes Menü
<F4>	neue Textweite
<F5>	Dokument zeigen
<F6>	Dokument weiter zeigen
<F7>	Tabulator vorwärts
<F8>	Hilfsbildschirm

Alle Tastenfunktionen von Superscript 128 auf einen Blick

Superscript 128 mit Star LC-10C

Default-Datei »Epson Typ Star LC-10C«:

Steuercodes	Funktion
40:	Textbreite für Bildschirm
4:	Druckernummer (0 = Centronics, 2 = RS232, 4 = seriell)
1:	Sekundäradresse für Normaldruck (255 = keine, z.B. bei Centronics-Anschluß!)
5:	Daten- und Stop-Bits für RS232
6:	Baud-Rate
1:	Parität
1:	1 = Zeilenvorschub wird erwartet, 0 = nicht
0:	1 = CBM-Codes werden erwartet, 0 = nicht
0:	1 = Cursor-Down-Mode, 0 = nein
0:	1 = Diablo-Codes, 0 = nein,
0:	1 = Spinwriter-Codes, 0 = nein
1:	1 = Breitschrift ein, 0 = aus
0:	1 = Drucker bringt Fettschrift per Backspace, 0 = nein
0:	1 = Drucker ist fähig zum Doppeldruck in 1/120-Inch-Schritten, 0 = nein
0:	1 = Unterstrich per Backspace, 0 = nein
2:	Anzahl, wie oft die Zeile in Fettschrift gedruckt wird
0:	1 = falls folgende Steuer-Bytes im CBM-Code interpretiert werden sollen
27, 45, 1:	Unterstreichen ein
27, 69:	Fettdruck ein (Emphasis)
27, 71:	Doppeldruck ein (Double)
15:	Schmalschrift ein
14:	Breitschrift ein
27, 83, 0:	Hochstellen ein
27, 83, 1:	Tiefstellen ein
27, 45, 0:	Unterstreichen aus
27, 70:	Fettdruck aus
27, 72:	Doppeldruck aus
18:	Schmalschift aus
20:	Breitschrift aus
27, 84:	Hochstellen aus (Zeilenvorschub)
27, 84:	Tiefstelle aus (negativer Zeilenvorschub)
27, 82, 0:	Sonderwunsch 1 (Zeichensatz USA)
27, 82, 3:	Sonderwunsch 2 (Zeichensatz UK)
27, 82, 7:	Sonderwunsch 3 (Zeichensatz Spanien)
27, 82, 1:	Sonderwunsch 4 (Zeichensatz Frankreich)
27, 82, 2:	Sonderwunsch 5 (Zeichensatz Deutschland)
27, 82, 6:	Sonderwunsch 6 (Zeichensatz Italien)
27, 52:	Sonderwunsch 7 Kursiv (= Italic) ein
27, 53:	Sonderwunsch 8 Kursiv aus
27, 112, 1:	Sonderwunsch 9 Proportionalschrift ein
27, 112, 0:	Proportionalschrift aus
:	setze 8 Zeichen/Inch
27 ,80:	setze 10 Zeichen/Inch
27, 77:	setze 12 Zeichen/Inch
:	setze 15 Zeichen/Inch
:	setze 20 Zeichen/Inch
27, 65, 18:	setze 4 Zeilen/Inch
27, 65, 12:	setze 6 Zeilen/Inch
27, 65, 9:	setze 8 Zeilen/Inch
27, 65, 6:	setze 12 Zeilen/Inch
27, 64:	Drucker-Initialisierung (wichtig, sonst unkorrekte Tastaturbelegung!)
35:	Nummernzeichen (Raute)
36:	Dollar
64:	§
91:	Ä
35:	
93:	Ü
94:	
123:	
124:	
125:	

126:	
16, 123:	ä
17, 124:	ö
18, 125:	ü
19, 126:	ß
22, 91:	Ä
23, 92:	Ö
24, 93:	Ü

Die Daten fürs serielle Treiberprogramm des Star LC-10C

Druckeinstellungen

Jetzt wird's bunt!

Textverarbeitungsprogramme verwenden individuelle Druckertreiber – keiner jedoch berücksichtigt Farbdrucker.

Schließen Sie Ihren Star LC-10 Color an den C128 an, aktivieren Sie ihn, laden und starten Sie das Programm mit: `RUN "LC-10 COLOR"`

Nach dem Startbild geht's per Tastendruck ins Hauptmenü (Abb.). Das Utility merkt automatisch, ob der Drucker aktiv ist und fordert Sie gegebenenfalls auf, ihn einzuschalten.

Eine Vielfalt der unterschiedlichsten Einstellmöglichkeiten steht Ihnen jetzt zur Verfügung:

- Farbwechsel,
- Zeilenabstand,
- Zeichensatz wählen,
- Fontgröße,
- Grafikdruck in verschiedenen Größen,
- Textformat (z.B. linker und rechter Rand),
- Aktivierung der Schriftarten Courier, Sanserif, Orator, NLQ und Draft.

Alle Optionen wählt man per entsprechender Zifferntaste, die <9> bringt Sie stets wieder ins Hauptmenü zurück.

Die gewünschte Einstellung wird vom Programm per Escape-Sequenz = CHR$(27) plus Steuer-Bytes an den Drucker weitergegeben. Sind Sie mit Ihrer Einstellung zufrieden, kann man per Taste <9> im Hauptmenü das Programm verlassen. Man hat nun die Möglichkeit, die eingetragenen Werte zu überprüfen (Option <8> Prüftext), ins Basic 7.0 zurückzukehren oder diverse C-128-Software von der entsprechenden Disk zu laden. Beachten Sie aber, daß einige Programme eigene Treiber verwenden, die Ihre neudefinierte Druckeinstellung teilweise ändern (z.B. Geos).

Sämtliche Befehlscodes stammen aus dem Handbuch zum Star LC-10 Color. Sie sind kompatibel zum Epson LX-800.

(Frank Steiner/bl)

Simple Druckeinstellung per Tastendruck

Kurzinfo: LC-10 Color

Programmart: Drucker-Installation
Bildschirmmodus: 80 Zeichen (VDC)
Laden und starten: RUN "LC-10 COLOR"
Besonderheiten: Der Drucker wird vor dem Start gewünschter Anwendungssoftware auf Farbbetrieb eingestellt (statt integriertem Druckertreiber)
Benötigte Blocks: 84
Programmautor: Frank Steiner

Nutzen Sie die Systemroutinen!

POKE, PEEK & Co.

Das Betriebssystem des C 128 nimmt Ihnen eine Menge Programmierarbeit ab, wenn Sie die eingebauten Assembler-Routinen aktivieren.

Eine Liste aller interessanten Speicherstellen und Routinenadressen des C128 würde ein Buch von mehreren hundert Seiten füllen. Hier also nur eine kleine Auswahl:

Befehl	Funktion
POKE 774,38: POKE775,160	nur Zeilennummern sichtbar
POKE 808,PEEK(808)-3	<RUN STOP/RESTORE> gesperrt
POKE 818,61: POKE 819,255	Reset nach SAVE-Befehl
POKE 792,0: POKE 793,64:	Einschaltbild per RESTORE-Taste
POKE 792,75: POKE 793,226	C-64-Modus per <RESTORE>
POKE 792,0: POKE 793,224	Reset bei <RESTORE>
POKE 820,189	Die ESCAPE-Taste ist jetzt im Quote-Modus (Anführungszeichen) sichtbar
POKE 808,112	<RUN STOP> aus
POKE 808,110	<RUN STOP> wirkt normal
POKE 808,225	Totalabsturz des C 128 (da hilft nur noch Ausschalten)
POKE 0,PEEK(0) OR 64: POKE 1,PEEK(1) AND 191	aktiviert den DIN-Zeichensatz ohne Drücken der entsprechenden Taste
POKE 0,47	stellt wieder den ASCII-Modus ein
POKE 2757,129	gibt im DIN-Modus alle Tastatureingaben nur in Großschrift aus; Zifferntasten bleiben er halten
POKE 2757,128	sperrt im 80-Zeichen-Modus die ASCII/DIN-Umschaltung
SYS 65366 oder 65363,0,8:	BOOT-Anweisung aktivieren (erste Zahl: SYS Laufwerksnr., zweite Zahl: Geräteadresse)
SYS 49194	Umschaltung zwischen 40- und 80-Zeichen modus
BANK 15: SYS 4352	Reset
SYS 65357 oder SYS 57931	C-64-Modus ohne Kontrollabfrage einschalten
POKE 1+256*PEEK(46),1: RENUMBER	stellt Basic-Programm nach NEW oder einem Reset wieder her
SYS 65378	VDC-Standardzeichensatz laden
POKE 981, banknummer	gewünschte Speicherbank setzen oder lesen
POKE 981,128	jeder SYS-Befehl wird falsch adressiert
PEEK(65)+256*PEEK(66)	aktuelle Zeilennummer für READ DATA
POKE 842,x bis POKE 851,x	Tastaturpuffer
PEEK(208)	Anzahl der Zeichen im Tastaturpuffer
POKE 2595,2	vergrößert den Tastaturpuffer auf 20 Speicherstellen
POKE 2595,0	Tastenpuffer ausschalten
POKE 2593,1	ersetzt GETKEY
POKE 243,1	revers an
POKE 243,0	revers aus
POKE 4566, low bis POKE4581, low	Low-Bytes der x- und y-Koordinaten (abwechselnd) der acht Sprites
POKE 4582, high bis POKE 4597, high	...die High-Bytes
POKE 902,128	erzeugt bei allen Befehlseingaben die Meldung »Syntax Error«
POKE 900,20	ab sofort zeigt kein Befehl mehr die gewünschte Wirkung
POKE 244,0	stellt im Programm-/RUN-Modus den Quote-Modus ab (entspricht der Tastenkombination <ESC O>).
POKE 245,64	sperrt Umschaltmöglichkeit von Groß- zu Kleinschrift
POKE 245,0	aktiviert wieder den Normalzustand
POKE 248,128	Bildschirm-Scrollen abstellen
POKE 248,0	...und wieder an
SYS 51602	erzeugt einen Ton (wie CHR$(7))
SYS 19910	entspricht der Anweisung END (ohne READY)
POKE 842,34: POKE 843,20: POKE 208,2	INPUT-Anweisung mit Komma
WINDOW 25,5,70,20,1: POKE 229,PEEK(229)+1: DIRECTORY	Directory mit fixierter Kopfzeile

Acorn Archimedes – neue Computer-Generation

Aufbruchstimmung

Muß es denn unbedingt ein IBM-kompatibler PC/AT sein? Wer den idealen Aufsteiger-Computer sucht, bekommt mit dem Archimedes von Acorn einen wahnsinnig schnellen 32-Bit-Rechner, der sich genauso leicht bedienen läßt wie der C 128.



[1] RISC-Power zum Dumping-Preis: A 3000

Mal ehrlich: Auch Sie haben schon mit dem Gedanken gespielt, Ihren treuen, aber technologisch angegrauten C128 durch einen superschnellen High-Tech-Rechner zu ersetzen. Als Alternative boten sich von jeher MS-DOS-Rechner, Amiga, Atari, Macintosh usw. an. Gewaltig im Kommen aber ist der Archimedes aus Großbritannien, der sämtliche Vorzüge der genannten Rechnertypen in sich vereint.

Diesen Computer steuert eine Zentraleinheit (CPU), die in puncto Geschwindigkeit alle anderen alt aussehen läßt: die RISC-Prozessoren ARM 2 bzw. 3, waschechte 32-Bit-Chips.

RISC – was ist das?

RISC ist die Abkürzung für »Reduced Instruction Set Code«. CPUs normaler Prägung (z.B. der Intel 80386 im AT oder der Motorola 68xxx im Amiga, Atari und Mac) besitzen zwar einen aufwendigen Befehlssatz; die auf den Rechnern eingesetzte Software beschränkt sich allerdings fast immer auf eine kleine Befehlsgruppe, die ständig wiederholt wird: Die übrigen, überflüssigen Anweisungen bremsen die Arbeitsgeschwindigkeit. Die RISC-Technologie faßt die wenigen, aber häufig benutzten Kommandos zusammen und stimmt die Hardware (= Prozessor, I/O-Chips, Floppy-Controller usw.) darauf ab. Damit nicht genug: Im Gegensatz zu den anderen CPUs benutzt RISC die Pipelining-Technik. Normale Prozessoren machen's nicht anders als die CPU 8502 des C128: Maschinenbefehl abholen (fetch), übersetzen (decode) und ausführen (execute) – schön eins nach dem anderen. Der ARM-Prozessor kann's aber gleichzeitig: Während die erste Instruktion ausgeführt wird, dekodiert er die zweite und holt bereits die dritte – auch eine Form von Multitasking! Damit (und durch die Fähigkeit, mit echten 32-Bit-Werten ohne Zwischenschritt zu rechnen) sind RISC-Computer die derzeit schnellsten auf dem Computer-Markt. Deshalb mißt man die



[2] Mit erweitertem Prozessor und Betriebssystem: A 3010

Geschwindigkeit von RISC-Prozessoren schon längst nicht mehr in MHz, sondern in MIPS (= million instructions per second). Der A 3000 ist z.B. dreimal schneller als ein 80386er und sechsmal so schnell wie der Amiga 600. Intel, Entwickler der 80xxx-Prozessoren, hat bereits einen eigenen RISC-Chip (Pentium, vorgestellt auf der CeBIT '93).

Äußerlich ähnelt der A 3000 (Abb. 1) dem Single-C-128 (eher noch dem Amiga 500): Mit der Zentraleinheit ARM 2 und dem Betriebssystem RISC OS 2.0 ist er bereits für 999 Mark zu haben, der modern gestylte A 3010 (Abb. 2), standardmäßig ausgerüstet mit dem erweiterten Prozessor ARM 250, RISC OS 3.1 und dem gleichen High-Density-Laufwerk wie der A 5000, kostet ca. 500 Mark mehr. Apropos: Fast genauso teuer war der 8-Bit-Computer C128 mit 122 KByte Speicher, als er 1985 auf dem deutschen Markt erschien – wohlgemerkt, ohne Floppy-Laufwerk! Da bietet der 32-Bit-Archimedes schon erheblich mehr: Multitasking und exakt 1024 KByte (= 1 Megabyte) freies Arbeits-RAM (aufrüstbar bis 4 MByte). Das Betriebssystem RISC OS befindet sich vollständig im ROM, ebenso die Benutzeroberfläche (Desktop, Geos-128-Freaks wird's freuen) und der integrierte Basic-Interpreter (Basic V): Die gewünschte Konfiguration (per eingebautem CMOS-RAM beliebig einstellbar) ist – wie vom C128 gewohnt – sofort nach dem Einschalten aktiv und muß nicht erst umständlich und zeitaufwendig von Disk oder Festplatte geladen werden.

Am interessantesten für ehemalige C-128-Programmierer ist zweifellos das eingebaute Archimedes-Basic-V. Es gilt als der Welt schnellster Basic-Dialekt – kein Wunder bei der Unterstützung durch den RISC-Prozessor! Verglichen mit der Riesenfülle von Anweisungen und Funktionen für Grafik, Diskettenoperationen, Dateiverwaltung usw. kommt einem das Basic 7.0 des C128 – auch in der erweiterten Version Masterbasic (s. Diskette zu diesem Sonderheft) – mickrig vor. Grafik-Freaks können sich mit kinderleichten Basic-Anweisungen nach Herzenslust austoben; je nach entsprechender Monitor-Einstellung (MODE) lassen sich bis zu maximal 256 Farben gleichzeitig anzeigen.

Wer gleich in Maschinensprache einsteigen will, greift auf den integrierten In-line-ARM-Assembler zurück (er kommt mit knapp 25 Befehlen aus). Alle Systemroutinen des RISC OS (SWIs = Software-Interupt-Routinen) lassen sich in eigenen Programmen nutzen. Im Gegensatz zu C-64/C-128-Assemblern, die Zahlen als Adreßangaben verlangen, kann man die Routinen per Namen aktivieren – aus Basic-Programmen ebenfalls mit der vom C128 bekannten Anweisung SYS.

Archimedes-Computer sind ab Werk mit einem 3½-Zoll-Laufwerk ausgestattet. Der A 3000 verarbeitet 800-KByte-Disketten im Archimedes-Format; mit RISC OS 3.1 akzeptiert die Floppy auch PC/AT-DOS- oder Atari-Formate. Das Laufwerk des A 3010 bzw. A 5000 formatiert High-Densitiy-

Disketten (doppelte Speicherkapazität, 1,66 MByte bei Archimedes-, 1,44 MByte bei PC-DOS- und 720 KByte bei Atari-Scheiben).

Als echte Profimaschine kann man den A 5000 (mit ARM-3-Prozessor, Abb. 3) kaum noch als Home-Computer bezeichnen. Im Preis (ca. 3300 Mark) inbegriffen: RISC OS 3.1, 2 MByte RAM, 40-MByte-Festplatte und das erwähnte HD-3½-Zoll-Laufwerk. Die durchschnittliche Rechenleistung: 14 MIPS (im Vergleich: der A 3000 bringt's auf 4 bis 5 MIPS, der A 3010 auf 8). Vier Expansion-Slots stehen bereit, den Computer mit Hardware-Karten aufzurüsten.

Archimedes-Laufwerk liest Fremdformate

Wer partout nicht aufs PC-Feeling verzichten will, legt sich den PC-Software-Emulator V1.8 (ca. 250 Mark) oder gleich die externe »386 PC Expansion Card« zu: Damit laufen z.B. Windows, Word, Lotus 1-2-3, AutoCAD, Corel Draw oder Word Perfect auch auf dem Archimedes! Die PC-Erweiterungskarte läßt sich übrigens an alle Archimedes-Typen anschließen, nicht nur ans Flaggschiff A 5000.

Wer den Desktop des Archimedes benutzt, wird Windows ohnehin keine Träne nachweinen – beim Kauf erhält der Anwender gratis eine Grundausstattung nützlicher Software (Applications Disc 1 und 2):

- !Draw und !Paint, zwei Mal- und Zeichenprogramme,
- !Edit, eine komfortable Textverarbeitung mit unterschiedlichen Fonts,
- !Maestro, Sound-Editor für Musikliebhaber, der die acht Stereo-Stimmen zu Höchstleistungen animiert,
- jede Menge Parallel-Treiberprogramme für Matrix- und PostScript-Drucker, Spiele, Sound- und Grafik-Demos.

Das kommerzielle Software-Angebot des Archimedes kann sich sehen lassen: professionelle DTP-Programme, CAD- und Grafik-Tools, Datenbanken, Chart-Programme, Tabellenkalkulationen, Programmiersprachen (C, Pascal, Cobol, LISP), Action-Spiele usw. Nicht zu vergessen: Inzwischen gibt's bei vielen Anbietern massenhaft PD-Software (prallvoll gefüllte Disketten ab 5 bis 20 Mark) – wenn wundert's, schließlich ist der Archimedes ebenso lange auf dem deutschen Markt wie der C128.

Mehr Infos finden Sie in den Archimedes-Sonderheften 1 (Dezember 1992) und 2 (ab 19.3.'93 am Kiosk). (bl)



[3] Rasend schnell und offen konzipiert: A 5000

```
(F1)........UHRZEIT : [illegible]
(F3)........ALARM : ON                          1 2 3 4 5 6 7 8
(RETURN)....EIN : 00:00.0 ........ AUS : 00:00.0
............EIN : 00:00.0 ........ AUS : 00:00.0
............EIN : 00:00.0 ........ AUS : 00:00.0
............EIN : 00:00.0 ........ AUS : 00:00.0
............EIN : 00:00.0 ........ AUS : 00:00.0
............EIN : 00:00.0 ........ AUS : 00:00.0
(F5)........ZEITEN SPEICHERN
MUSIC MASTER by Christoph Güntner, Merowingerstr.13, 8032 Gräfelfing
```

Professionelle Datenbank für Musik-Fans: Music Master V7.70

Jetzt für serielle Drucker

Bytes, auf die's ankommt

Darauf haben viele gewartet: Unser Audio-Archiv für CDs, LPs und Musikkassetten im 128er-Sonderheft 76 arbeitet jetzt auch mit seriell angeschlossenen, Epson-kompatiblen Druckern!

Endlich ist es da: Das Patchprogramm zu »Music-Master V7.70« (Abb.) ändert die im Hauptprogramm integrierte Druckroutine. Ab sofort läßt sie sich auch mit Epson-kompatiblen 9-Nadel-Druckern einsetzen (nicht nur 24-Nadler!).

Unser Patch-Programm auf der Rückseite der beiliegenden Sonderheft-Disk macht's möglich. Laden und starten Sie mit:

```
RUN "PATCH MUSIC"
```

Das Programm fordert Sie auf, Ihre Systemdisk zu Music-Master ins Laufwerk zu legen (oder die Vorderseite der Diskette zum 128er-Sonderheft 76). Per Tastendruck lädt der Computer die Hauptdatei »C-Music Master« (230 Blocks) in Bank 1, ändert die Bytes im Speicher und schreibt das gepatchte Programm (nicht ohne die alte Version vorher gelöscht zu haben) wieder auf die Programm-Disk zurück.

Anschließend holt das Patch-Programm die neue Druckeranpassungs-Version wieder in den Speicher. Wenn die Meldung »Fertig« erscheint, sollten Sie diesen Treiber-Generator unbedingt mit RUN starten: um die entsprechenden Steuer-Bytes, ESC-Sequenzen und den passenden Treiber (Music Dat LC24, LC10 oder IBM) für Ihren seriellen Drucker einzustellen (dabei hilft Ihnen der Griff zum Druckerhandbuch). Sichern Sie die geänderte Datei wieder auf Ihrer Programm-Disk. (Christoph Güntner/bl)

Kurzinfo: Patch Music

Programmart: Patch-Datei
Bildschirmmodus: 80 Zeichen (VDC)
Laden und starten: RUN "PATCH MUSIC"
Besonderheiten: Nach der Änderung funktioniert Music-Master V7.70 auch mit seriell angeschlossenen 9-Nadel-Druckern
Benötigte Blocks: 9
Programmautor: Christoph Güntner

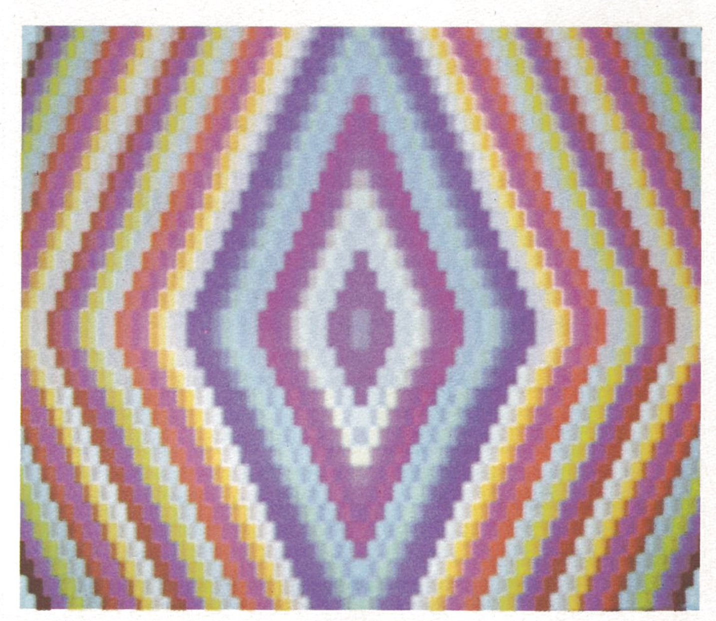
Grafikprogramm

Farbenpracht 128

Mit seinen 16 Grundfarben sieht der C 128 sehr trüb und blaß aus. Das Programm »Moving Colors 80« aber bringt gewaltig Farbe ins Spiel.

Das Programm »MOVING COLORS 80« arbeitet auf dem 80-Zeichen-Screen. Es benutzt den 64K-RAM-VDC. Nach dem Start bewegt sich ein Farbbildschirm (s. Abb.) und mit <SPACE> schaltet man auf das nächste Muster. Mit <SHIFT-LOCK> und der linken Shift-Taste läßt sich die Bewegung stoppen. Mit Hilfe des Pulse-Controlled-Colors-Verfahren (PCC) werden bis zu 136 flimmerfreie Farben auf dem 80-Zeichen-Bildschirm dargestellt. Bei dem Verfahren wird der Hires-Schirm eingeschalten und abwechselnd die Masken 01010101 in gerade Zeilen und 10101010 in die ungeraden Zeilen geschrieben (0 = Hintergrund, 1 = Vordergrund). Dadurch entstehen 136 Mischfarben. Die Bewegung wird durch ständige Erhöhung der Farbstelle erzielt. Gearbeitet wird mit drei Auflösungen (s. Tabelle). Die Muster erzeugt das Programm durch Vierecke mit verschiedenen Maßen. Die Maschinenroutine SYS 4867,rx,ry,ymax setzt diese Vierecke. Der Befehl SYS 4867,10,8,32 erzeugt beispielsweise vier Rechtecke mit 20 Pixeln und vertikal zwei Vierecke mit 16 Pixeln.

Zum Bewegen des Musters dient die Maschinenroutine SYS 4864,BL, wobei BL der Wert für das High-Byte vom Speicherende ist, der bewegt werden soll. Die Start-Adresse des Musters liegt bei $8000. Für 32 Farbzeilen liegt die Endadresse bei $8a00, BL bekommt hier den Wert 138 ($38). Für 64 Farbzeilen ist BL = 148 ($94) und bei 128 ist BL = 168 ($ab). Eigene Muster schreibt man in Bank 0 ab $8000. U ist die Variable für die Auflösung und kann 0,1 und 2 (32, 64, 128 Farbzeilen) betragen. Mit GOSUB 210 wird die Auflösung eingestellt und die Bewegung gestartet. In den REM-Zeilen 150 und 160 des Programms findet man ein Beispiel für eigene Muster. Die Routine in Zeile 150 füllt den Bereich $8000 bis $a800 (80 mal 128 Farbpunkte) mit einem Streifenmuster. Der Vorgang dauert ca. 50 Sekunden. Zeile 160 startet die Bewegung in drei verschiedenen Auflösungen. (P. Guldenaar/lb)



Viele Farben und Muster auch auf dem C128

Auflösungen bei »Moving Colors 80« (Tabelle)
80 mal 32 Farbpunkte
80 mal 64 Farbpunkte
80 mal 128 Farbpunkte

Kurzinfo: Moving Colors 80

Programmart: Grafikprogramm
Bildschirmmodus: 80 Zeichen (VDC)
Laden und Starten: RUN "MOVING COLORS 80"
Besonderheiten: erzeugt bis zu 136 Farben auf dem 80-Zeichen-Schirm
Benötigte Blocks: 6
Programmautor: Paul Guldenaar

Eingabe-Utility

Cursor mit Maus

Bei der Arbeit auf dem 80-Zeichen-Schirm dauert es oft sehr lange mit den Tasten den Cursor zu positionieren. Mit einer Maus oder einem Joystick geht das viel schneller...

Die Bewegung des Cursors mit den Tasten läuft sehr träge ab. Eine Hilfe ist da der Einsatz einer Maus. Mit dem Programm »Maus-Cursor« kann der Positionszeiger blitzschnell über den Bildschirm bewegt werden. Das Programm wird mit BLOAD geladen und mit SYS 2816 gestartet. Nun ist die Maus als Eingabegerät für den Cursor aktiv. Man kann die Steuerung problemlos in eigene Programme einbauen, muß aber einiges dabei beachten. Die Tabelle zeigt alle wichtigen Adressen, wobei zu beachten ist, daß das Programm ab 2816 ($0B00) bis 2962 ($0B92) im Kassettenpuffer liegt.

(M. Müller/lb)

Die Adressen von Maus-Cursor (Tabelle)

65320	$dc00	Port 1 für Maus bzw. Joystick
250	$00fa	Zähler für Geschwindigkeit der Abfrage
2838	$0b16	Geschwindigkeit der Bewegung
235	$00eb	Aktuelle Zeile
236	$00ec	Aktuelle Spalte

Kurzinfo: Maus-Cursor

Programmart: Eingabe-Hilfe
Bildschirmmodus: 80 Zeichen (VDC)
Laden: BLOAD "MAUSCURSOR"
Starten: nach dem Laden SYS 2816 eingeben
Besonderheiten: in eigene Programme integrierbar
Benötigte Blocks: 1
Programmautor: Mario Müller

Auto-Boot 128 – für die 1581!

Start frei für die Micro-Floppy

Fehler sind da, um ausgemerzt zu werden. Die neue Version des »Auto-Boot-Maker« von der Test-Demo-Disk zur 1581 hält jetzt endlich, was Commodore schon immer versprach.

Haben Sie schon versucht, mit dem Originalprogramm einen Boot-Sektor für Disketten im Laufwerk 9 zu installieren? Zunächst klappt alles nach Wunsch (Disk Unit Nr. 9, Programmname und Datentyp eintragen usw.), aber dann kommt der große Einbruch: Wenn man das definierte Programm per Anweisung BOOT U9 von der 1581 starten will, erscheint die Fehlermeldung »File not found«. Der Grund: Die Block Availability Map (BAM) der 1581 liegt in den Sektoren 1 und 2 der Spur 40, nicht in Spur 18, Sektor 0 (wie bei der Floppy 1571). Das kümmert aber das Commodore-Programm wenig - es greift ungeniert auf Laufwerk 8, Track 18, zu. Damit ist die Routine für 1581-Besitzer wertlos, die diese Floppy als Zweitlaufwerk mit Geräteadresse 9 benutzen:

Durch Änderung der Basic-Zeilen 107, 114 und Einfügen der neuen (Nr. 2120 bis 2124) läßt sich das Programm aufpeppen:

```
107 gosub 2120
114 print#15,"u1:8 0"t;s:print#15,"b-p";8;by
2120 print"1.) 5 1/4"chr$(34)" disk drive"
2121 print"2.) 3 1/2"chr$(34)" disk drive"
2122 input"choose drivetyp (1/2)";a$:if a$<>"1"
 and a$<>"2" then printchr$(145)
 chr$(145)chr$(145)chr$(27)chr$(64);:goto 2120
2123 if a$="1" then t=18: s=0: by=5
2124 if a$="2" then t=40: s=1: by=17
```

Zusätzlicher Trick: In Zeile 228 fragt das Programm nun nicht mehr nach »U(Geräteadresse)«, sondern nach »U(peek (186))« - der Speicherstelle, in der die Geräteadresse des aktuellen Geräts gespeichert ist.

- Ab sofort kann man neben den vier verschiedenen Geräteadressen auch festlegen, ob man den Boot-Sektor auf einer 5¼- oder 3½-Zoll-Disk etablieren will (Taste <1> oder <2>).
- BOOTet man jetzt von Laufwerk 9 (BOOT U9), wird das gewünschte Programm anstandslos geladen - als wär's die Floppy Nr. 8!

In »Auto-Boot 128« auf der Rückseite unserer Diskette zu diesem Sonderheft sind die erwähnten Änderungen bereits eingebaut. (Thorsten Oelfke/bl)

Kurzinfo: Auto-Boot 128

Programmart: Utility
Bildschirmmodus: 40- und 80 Zeichen
Laden: RUN "AUTO-BOOT 128"
Besonderheiten: Im Gegensatz zum Originalprogramm werden nun auch 1581-Laufwerke berücksichtigt!
Benötigte Blocks: 11
Programmautor: Thorsten Oelfke

DiskEtti-128-Update

Labels mit High-Speed

Noch schneller, noch komfortabler: Unser Druckprogramm für Diskettenaufkleber im 128er Sonderheft 76 erledigt jetzt seine Aufgabe blitzschnell!

Das Umrechnen der Grafikdaten und der Ausdruck der Labels ist in der alten Version äußerst zeitintensiv. Die Routinen wurden also in Assembler programmiert (»Disketti.MC«). Resultat: Das Installationsprogramm reduziert die Wartezeit von drei Minuten auf acht Sekunden, außerdem arbeitet die Druckroutine jetzt zehnmal schneller!

Die geänderte Programmversion des Software-Pakets finden Sie auf der Diskette zu diesem Sonderheft: DISKETTI I/NEW, DISKETTI M/NEW, DISKETTI.MC.

Neu ist die Assembler-Datei »Disketti.MC«, die nach dem Start von »Disketti Install« nachgeladen wird: Sie liegt im Speicherbereich von $1300 bis $13EF und enthält die entsprechenden Maschinensprache-Routinen zum Drucken.

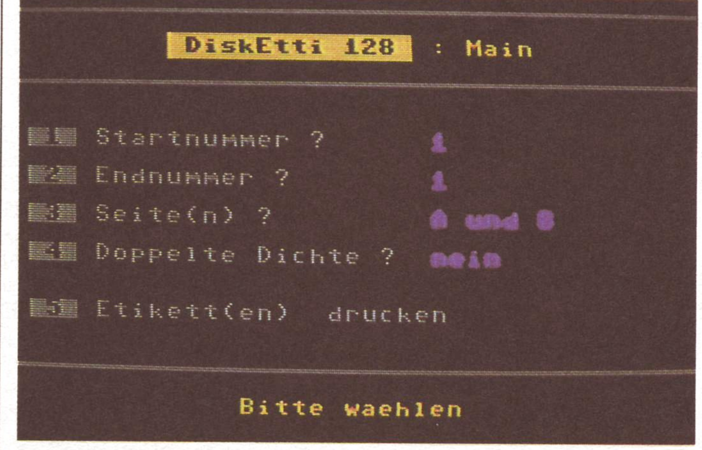
Laden und starten Sie mit:

```
RUN "DISKETTI I/NEW"
```

Tragen Sie die vorgesehenen Werte ein, speichern Sie die Daten (Filename: DISKETTI.D), verlassen Sie das Programm und aktivieren Sie anschließend das Hauptprogramm:

```
RUN "DISKETTI M/NEW"
```

An Bedienung und Programmfunktionen hat sich nichts geändert (s. 128er-Sonderheft 76, S. 12). (bl)



Per Installationsmenü gestaltet man Diskettenaufkleber

Kurzinfo: DiskEtti 128/II

Programmart: Disketten-Labels drucken
Bildschirmmodus: 40 Zeichen (VIC)
Laden: RUN "DISKETTI M/NEW"
Besonderheiten: Grafik- und Drucker-Routine jetzt als Assembler-Datei
Benötigte Blocks: 46
Programmautor: Michael Kübel

Floppy-Tool

Disk-Hilfen

Wer mit den Befehlen zum Disketten-Handling auf dem C 128 Probleme hat, ist mit »Disk-Hilfen« gut bedient.

Für viele User sind die Befehle zum Arbeiten mit dem Disk-Laufwerk zu umständlich. Für sie ist das Tool »Disk-Hilfen« die ideale Unterstützung. Die Hauptpunkte:

- Directory auf dem Drucker ausgeben
- Autoboot-Sektor installieren
- Disk-Inhalt bearbeiten
- Files-Handling

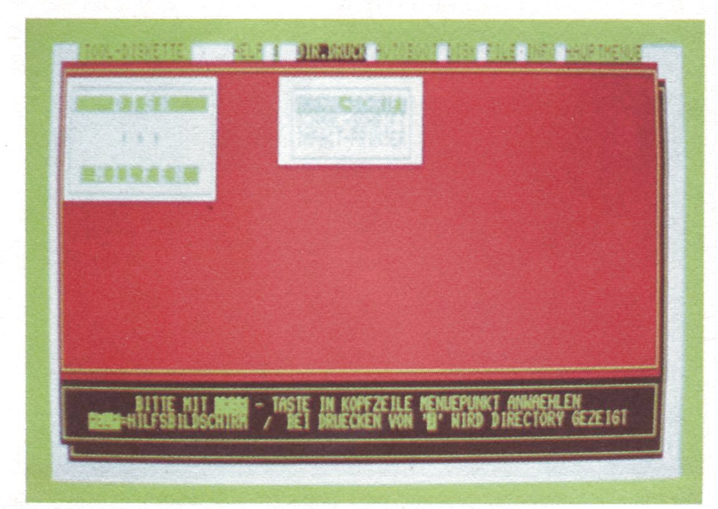
Um ein Inhaltsverzeichnis auf dem Drucker auszugeben, hat der Nutzer wiederum drei Möglichkeiten:

Bei Normalschrift wird es einspaltig ausgegeben, bei Schmalschrift in bis zu fünf Spalten und als dritte Möglichkeit, werden Impact-Drucker angesprochen.

Wer ein File autobootfähig machen möchte, kann dies sowohl mit Basic-, als auch mit Assembler-Programmen. Es wird nach der Wahl der Programmart, kurzerhand der Name des Files angegeben und der Computer besorgt den Rest.

Bei Einsatz von Disketten kann man solche formatieren, sie vor Schreibzugriffen schützen, den Schutz wieder aufheben, den Header und die ID verändern und mit »Aufräumen« offene Files von der Disk entfernen.

Im File-Menü kann man zahlreiche Manipulationen vornehmen. Die Files lassen sich umbenennen, kopieren, miteinander verbinden, löschen, gelöschte Files wieder herstellen, schützen und den Schutz wieder aufheben. Jede Aktion kann man mit der Escape-Taste abbrechen und wenn man detaillierte Infos zu den Funktionen haben will, benutzt man nur die Help-Taste. (Ulrich Bürsgens/lb)



Das Hauptmenü der Disk-Hilfe

Kurzinfo: Disk-Hilfe

Programmart: Disk-Tool
Bildschirmmodus: 80 Zeichen (VDC)
Laden und Starten: RUN "DISK-HILFEN"
Besonderheiten: Bedienungshilfen im Programm
Benötigte Blocks: 153
Programmautor: Ulrich Bürsgens

SONDER HEFT 90

VORSCHAU

64'er

DISKETTE IM HEFT

Wer spielt, hat mehr vom Leben! Wir denken dabei nicht an Lotto-Jünger oder Roulette-Fetischisten, sondern an alle ausgefuchsten Joystick-Zocker unter den Zigtausenden C-64-Fans. Unser nächstes Spiele-Sonderheft (Nr. 90) bietet wieder eine raffinierte Auswahl superstarker Action-, Adventure und Knobel-Games:

- Bei »Cyborg 2900«, dem Joystick-Abenteuerspiel mit 3-D-Grafik taucht man hautnah ins Geschehen ein,
- Der Klassiker »Tetris« stand Pate bei »Colors«. Die ausgefeilte Grafik schlägt die Originalversion aber um Längen!
- Mit unserem »Boulder Dash-Construction-Editor« entwickeln Sie brandneue Levels für den Diamantensucher Rockford und bauen sie problemlos in diesen Spiele-Evergreen ein.
- Haufenweise Tips, Tricks und jede Menge Cheatmodi sorgen dafür, daß man vom höchsten Level inkl. Endmonster nicht lange träumen muß!

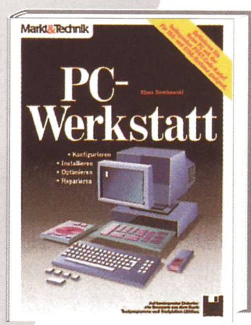
Aus aktuellen oder technischen Gründen können Themen ausgetauscht bzw. verschoben werden. Wir bitten um Ihr Verständnis.

Nr. 90 gibt's ab 27.05.93 bei Ihrem Zeitschriftenhändler

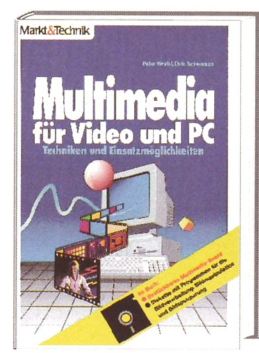
Sichern Sie sich professionelles HARDWARE-WISSEN!

An alle Hardware-Einsteiger und Hardware-Profis: Die Experten des Markt&Technik-Verlages bieten Ihnen ihr Wissen an. Grundlegend und umfassend - professionell und verständlich. Vom Standard-Werk der Hardware über ein aufregendes Multimedia-Buch bis hin zur detaillierten Anleitung zum Aufbau eines Mini-Rechners. Sichern Sie sich genau das Hardware-Wissen, das Sie brauchen. Denn nur wer die Hardware seines PCs kennt, kann wirklich effizient arbeiten!

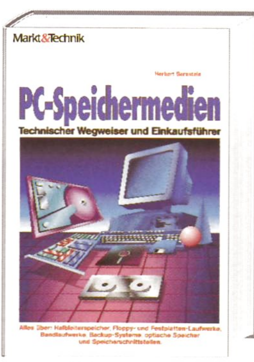
Holen Sie sich wertvolles Hardware-Wissen: Buch + Diskette + Platine!



Klaus Dembowski
PC-WERKSTATT
Der Leser erfährt hier alles über die Systemkomponenten des PC, so daß jede Scheu vor dem Kontakt mit der Hardware entfällt. Speicheraufrüstung oder ein zusätzliches Laufwerk sind damit kein Problem mehr. Und wenn der Rechner überhaupt nicht mehr funktioniert, kann man sich mit Hilfe der beiliegenden Platine eine Post-Code-Karte zur Fehlerdiagnose selber bauen.
1992, ca. 300 Seiten, inkl. Diskette und Platine
ISBN 3-877**91-344**-X ca. DM 98,-



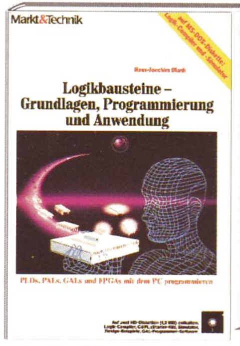
P. Wratil/D. Schwampe
MULTIMEDIA FÜR VIDEO UND PC
Einladung zur Entdeckungsreise ins Multimedia-Land. Wie Sie die aufregenden Möglichkeiten der Technik selbst erleben können, beschreiben die Autoren in diesem Praxisbuch, das eine unbestückte Platine enthält. Mit etwas Lötzinn und Erfahrung läßt sich daraus eine Genlock- und Multimedia-Karte basteln. Ein Multimedia-Software-Paket und Diagnose-Software werden auf Diskette mitgeliefert.
1992, 372 Seiten, inkl. Diskette und Platine
ISBN 3-877**91-194**-3 DM 98,-



Herbert Bernstein
PC-SPEICHERMEDIEN
Alles über Halbleiterspeicher, Floppy- und Festplatten-Laufwerke, Backup-Systeme, optische Speicher und deren Schnittstellen. Ein technischer Wegweiser und Einkaufsführer für fortgeschrittene PC-Anwender, die ihren Computer eigenhändig erweitern wollen. Neben ganz praktischen Kauf- und Einbau-Tips erfährt der Leser eine Menge über die Funktion der Bauteile.
1992, ca. 800 Seiten, inkl. Diskette
ISBN 3-877**91-162**-5 ca. DM 89,-



Klaus Dembowski
PC-GESTEUERTE MESSTECHNIK
Die praxisgerechte Anleitung zur Entwicklung von Meßsystemen mit Hilfe von Einsteckkarten, der RS232- und der IEC-Schnittstelle. Einsteigern und Profis liefert dieses „Kochbuch" neben erprobten Konzepten und Hintergrundwissen als Besonderheit eine IEC-Bus-Platine, die voll kompatibel zum Industriestandard ist. Die Software für die Platine und alle im Buch beschriebenen Meßprogramme werden auf einer Diskette mitgeliefert.
1991, 471 Seiten, inkl. Diskette und Platine
ISBN 3-890**90-958**-2 DM 119,-



Hans-Joachim Blank
LOGIKBAUSTEINE - GRUNDLAGEN, PROGRAMMIERUNG UND ANWENDUNG
Dieses vielseitige Praxisbuch beschreibt vor allem die Programmierung der modernen Logikbausteine. Und es stellt viele interessante Anwendungsbeispiele vor, z.B. Paßwortdecoder, Aufzugssteuerung oder Multibus-Schnittstelle. Auf zwei HD-Disketten (5,25") werden als Starter-Kit mitgeliefert: Logik-Compiler, Simulator für die Bausteinfunktionen, GAL-Programmer-Software und Designbeispiele!
1992, 421 Seiten, inkl. 2 Disketten
ISBN 3-877**91-072**-6 DM 79,-

Außerdem lieferbar:

Herbert Bernstein • **HARDWARE-HANDBUCH FÜR PC/XT/AT UND KOMPATIBLE** • 1990, 431 Seiten • ISBN 3-89090-913- 2 DM 79,-

J. Koch/M. Schusser • **PC/XT/AT-KOMPENDIUM** • 1989, 504 Seiten, inkl. Diskette • ISBN 3-89090-778-4 DM 69,-

Kai Hamann • **PC-BASTELBUCH** • 1990, 309 Seiten, inkl. Diskette und Platine ISBN 3-89090-331-2 DM 98,-

Herbert Bernstein • **PC-TUNING** • 1991, 552 Seiten, inkl. Diskette ISBN 3-89090-950-7 DM 69,-

Uwe Gerlach • **DAS TRANSPUTERBUCH** • 1991, 464 Seiten, inkl. Diskette und Platine • ISBN 3-87791-019-X DM 119,-

P. Wratil/R. Schmidt • **DER PC ALS INTELLIGENTE SCHALTZENTRALE** 1990, 506 Seiten, inkl. Diskette und Platine • ISBN 3-89090-651-6 DM 119,-

H.-J. Blank/H. Bernstein • **PC-SCHALTUNGSTECHNIK IN DER PRAXIS** 1990, 506 Seiten, inkl. Diskette und Platine • ISBN 3-89090-914-0 DM 119,-

P. Wratil/R. Schmidt • **PC/XT/AT - MESSEN, STEUERN, REGELN** 1987, 255 Seiten, inkl. Platine • ISBN 3-89090-477-7 DM 99,-



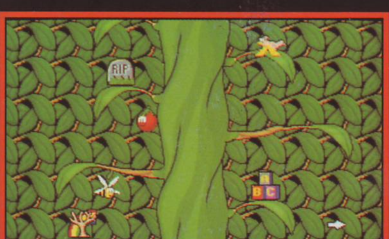
DAS ERFOLGS-PROGRAMM FÜR IHR PROGRAMM!

Dies ist nur ein kleiner Ausschnitt aus dem neuen Gesamtprogramm des Markt&Technik-Verlages: Mehr als 500 Problemlösungen zu Hard- und Software warten auf Sie - jetzt bei Ihrem Buchhändler, im PC-Fachhandel und in den Computer-Abteilungen der Warenhäuser!

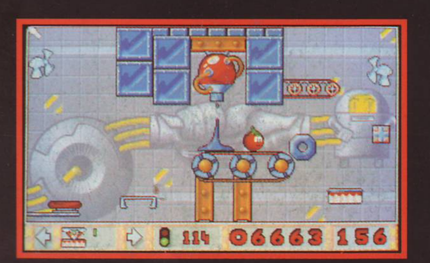
6221-1

Bill's TOMATO GAME™

Bunter als ein Picasso!
Süßer als Schokoladeneis!!
Frecher als alle anderen Früchtchen!!!
Machen Sie die Bekanntschaft von T'n'T, einem ganz gewöhnlichen Tomatenpärchen, und erleben Sie eine explosive Mischung von wahrer Liebe und (ent) fesselnden Bildschirm-Orgien mit mehreren Ventilatoren.
Ketch-up!











Deutsche Vertretung: Alliance Division, Verlerstr. 1, 4830 Gütersloh